10 1990

P.1877/90

informatyka

Grafika i Widzenie

WYDAWNICTWO NOT SIGMA

Cena 5800 zł

informatyka

nr 10 1990
Październik
Rok wydania XXV

Miesięcznik

ISSN 0542-9951. INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
dr inż. Wacław ISZKOWSKI
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Jan RYŻKO
mgr Hanna WŁODARSKA

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA: Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4,
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja: 01-552 Warszawa,
Pl. Inwalidów 10 p. 128, 136
tel. 39-14-34

Materiałów nie zamówionych redakcja nie zwraca

DRUK:
PRASOWE ZAKŁADY GRAFICZNE
RSW „PRASA-KSIĄŻKA-RUCH"
ul. Dworcowa 13,
85-950 BYDGOSZCZ
zam. 2445/90
Obj. 4,0 ark. druk. Nakład 5250 egz.

Cena egzemplarza 5800 zł

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji

W numerze:

	Strona
Wprowadzenie do maszynowego postrzegania – *Wojciech Mokrzycki*	1
Mikrokomputerowe systemy przetwarzania obrazów – *Andrzej Materka, Włodzimierz Górnisiewicz, Mariusz Kukula*	9
Problemy współbieżnej rekonstrukcji obrazów – *Wiesław Lucjan Nowiński*	19
Zastosowanie grafów kompozycyjnych do konstrukcji strukturalnego edytora graficznego – *Ewa Grabska, Grażyna Hliniak, Jacek Lembas*	23
/unix	
Jak startuje UNIX – *Dariusz Kupiecki*	28
Systemy otwarte i ich wrogowie, czyli co dalej z UNIX-em (2). Główni konkurenci – *Jacek Surma*	29
Kosz na śmieci – *Władysław Majerski*	30
zBITki	
IF Informatyk THEN Kto? ELSE Programista – *Wacław Iszkowski*	III okł.

W najbliższych numerach:

- Jarosław Deminet o komputerach w Kongresie USA.
- Wojciech Skaba o współczesnych metodach kompresji danych – o teorii, algorytmach i zastosowaniach.
- Katarzyna Lange o zasadach działania i przykładach zastosowań poczty elektronicznej w sieci ARCNET.
- Krzysztof Dworakowski i in. o podstawowych cechach oprogramowania systemu zarządzania hierarchiczną bazą danych GRBASE.

Warunki prenumeraty

Przyjęcie prenumeraty – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorom przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

Blankiet wpłaty – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

Wpłata – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj.:
Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11

Prenumeratorzy zbiorowi – osoby prawne obowiązują blankiety „Wpłata-Zamówienie". Cena normalna.

Prenumeratorzy indywidualni – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

Prenumerata ulgowa – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura".
Uwaga: w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

Prenumerata ze zleceniem wysyłki za granicę – cena prenumeraty jest dwukrotnie wyższa od ceny normalnej.
Należy podać dokładny adres odbiorcy za granicą.

Terminy przyjmowania prenumeraty:
do 10 listopada na I, II, III, IV kwartał następnego roku
do 28 lutego na II, III, IV kwartał br.
do 31 maja na III, IV kwartał br.
do 31 sierpnia na IV kwartał br.
Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

Egzemplarze archiwalne (z lat ubiegłych)
Można nabyć za gotówkę w Klubie Prasy Technicznej w Warszawie, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa skr. poczt. 1004 (tel. 40-37-31) na rachunek lub za zaliczeniem pocztowym.

Informacji o prenumeracie udziela: Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartycka 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

Wstępna cena jednego egzemplarza na 1990 rok: normalna – 5800 zł
ulgowa – 2900 zł.

Wartość prenumeraty (w zł):
Normalna: kwartalna – 17 400, półroczna 34 800, roczna 69 600
Ulgowa: kwartalna – 8 480, półroczna 6 960, roczna 13 920

Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratorzy zobowiązani są do dopłaty różnicy cen.

IF Informatyk THEN Kto? ELSE Programista

Kto to jest informatyk? Dyskusja nad definicją zawodu określanego hasłem „informatyk" toczy się od wielu lat i rodzi wiele emocji.

Według Statutu Polskiego Towarzystwa Informatycznego, członkami zwyczajnymi Towarzystwa mogą być obywatele polscy, którzy:
1) ukończyli studia wyższe na kierunku informatycznym lub związanym z informatyką lub mają stopień naukowy w zakresie informatyki lub jej zastosowań;
2) mają wykształcenie wyższe lub średnie, a ich praca zawodowa w ciągu co najmniej trzech ostatnich lat przed wstąpieniem do Towarzystwa była ściśle związana z informatyką;
3) studiują na kierunkach informatycznych lub związanych z informatyką, poczynając od trzeciego roku studiów.

Punkt 2. rodzi jednak czasem wątpliwości, na ile zeprezentowany kontakt z informatyką ma rzeczywisty związek z nabyciem umiejętności tworzących wiedzę w tej dziedzinie. Niestety brak jest tutaj jednoznacznych odpowiedzi, gdyż inaczej widzą tę sprawę informatycy po studiach, a inaczej wszyscy ci, którzy zmienili swoje zainteresowania pod wpływem możliwości dostępu do komputera.

Po pierwsze, trzeba sobie uświadomić fakt, że jeszcze większość zawodowych informatyków wyrastała łącznie z początkowym rozwojem tej dziedziny, mając w tym rozwoju znaczny udział. Większość kadry ze stopniami naukowymi kończyła inne warunki studiów. Dopiero od połowy lat 70. absolwenci uczelni uzyskiwali na dyplomie potwierdzenie ukończenia specjalności „informatyka". Jednocześnie gwałtowny rozwój informatyki oraz rozpowszechnienie mikrokomputerów wymagały coraz większej liczby kadr, znacznie przekraczającej możliwości polskich uczelni.

Tak duże zapotrzebowanie na specjalistów potrafiących eksploatować i oprogramowywać komputery jest nie tylko naszym krajowym problemem. Np. w USA istnieje opinia, że obecnie, a także w najbliższej przyszłości, potrzeba będzie znacznej liczby programistów. W niektórych prognozach mówi się nawet, że liczba ta może dorównać liczbie ludności tego kraju. Trzeba jednak wyjaśnić, że nie każdy programista jest tam utożsamiany z informatykiem. Na programistów szkoli się osoby zajmujące się poprzednio zupełnie czymś innym. Programiści tacy w wielu przypadkach znają się tylko na jednym systemie operacyjnym i potrafią programować – na przykład – tylko w Cobolu. Często też są oni zatrudniani czasowo na kontraktach organizowanych przez firmy konsultatywne.

Sytuacja ta przypomina nieco początki rozwoju motoryzacji, kiedy to do każdego nowego automobilu musiał być dodany kierowca – mechanik. Oczywiście z czasem kierownice przejęli sami właściciele, a następnie ich żony. Obecnie szkoli się tylko techników mechaników potrafiących dokonać naprawy. Nikt ich nie utożsamia z inżynierami mechanikami projektującymi nowe auta, którzy obecnie korzystają z zaawansowanych technik komputerowych. Zamiast tuneli aerodynamicznych wykorzystuje się technikę symulacyjną, a kształty nadwozi opisują równania matematyczne zamiast drewnianego kopyta. I nie sądzę, aby któryś z tych inżynierów uważał się za informatyka.

Teraz mogę przystąpić do podania mojej definicji informatyka, przy czym rozgraniczę tutaj inżyniera informatyki o specjalizacji sprzętowej od informatyka o specjalizacji w zakresie oprogramowania.

Inżynier informatyk o specjalizacji sprzętowej powinien rozumieć organizację funkcjonalną sprzętu komputerowego, umieć konstruować oraz naprawiać ten sprzęt przy poważniejszych uszkodzeniach. Jego podstawową rolą jest opracowywanie nowych rozwiązań sprzętowych opartych na nowych możliwościach technologicznych w zakresie układów scalonych oraz techniki zapisu magnetycznego, optycznego itp. Patrząc na te wymagania, możemy na naszym podwórku tylko pokiwać głową. Jednak, kiedy zbliżymy się do tego nowocześniejszego świata nie powinniśmy zaniechać poznawania tej wiedzy, aby ją rozumieć i wdrażać w kraju.

Pozostaje też szerokie pole popisu w konstruowaniu specjalizowanych sprzętowych urządzeń informatyki, uzupełniających standardowe instalacje.

Informatyk o specjalizacji w zakresie oprogramowania (zarówno inżynier, jak i absolwent uniwersytetu) powinien rozumieć architektury nowoczesnych komputerów oraz posiadać umiejętność poznawania zasad funkcjonowania działającego na nich oprogramowania. Powinien on umieć samodzielnie opracowywać rozwiązania w zakresie tworzenia nowego oprogramowania systemowego, translatorów oraz nowych języków programowania i pakietów przetwarzania danych. Zadaniem takiego informatyka jest zrozumienie potrzeb potencjalnych użytkowników, wskazywanie im odpowiedniego sprzętu, a także dobieranie lub konstruowanie odpowiedniego oprogramowania podstawowego i wspomagającego. Powinien też móc porozumieć się z użytkownikiem w jego języku rozumienia problemu, sugerując mu najlepsze drogi postępowania. Jego głównym zadaniem jest nauczenie użytkowników komputerów sposobu ich wykorzystywania.

Może on też brać udział w tworzeniu aplikacji, wskazując na rozwiązania najefektywniej wykorzystujące sprzęt oraz istniejące już oprogramowanie. Takie postawienie zagadnienia prowadzi do wniosku, że o jakości informatyka nie decyduje liczba napisanych przez nich programów. Może on wręcz w ogóle nie pisać programów aplikacyjnych, gdyż w wielu

dokończenie na s. 17

PRZEDSIĘBIORSTWO ZASTOSOWAŃ INFORMATYKI

meditronik

oferuje:

- systemy mikrokomputerowe
- programy aplikacyjne dla różnych dziedzin gospodarki (na życzenie wysyłamy katalog)
- poszukiwane komponenty elektroniczne
- interfejs do kamery video (opc. CCD) z bogatą biblioteką oprogramowania
- emulator Z80
- tester układów scalonych i pamięci
- programator BPROM
- asynchroniczny procesor komunikacyjny
- konwerter RS-232 – Centronics

instaluje:

- połączenia międzykomputerowe (XT/AT – ODRA/RIAD/IBM)
- systemy sieciowe (NOVELL)
- systemy wielodostępne (SCO Xenix 286, 386, Unix System V)

Jeżeli jesteś autorem oryginalnego programu aplikacyjnego
- skontaktuj się z nami, będziemy pośredniczyć w sprzedaży Twojego programu, dbając o ochronę Twoich praw autorskich!

Nasz adres:

00-194 Warszawa, ul. Dzika 4
tel. (02) 635-22-63, 635-22-64,
fax (02) 635-21-95
tlx 816075 medi pl

EO/605/89

WOJCIECH MOKRZYCKI
Instytut Podstaw Informatyki PAN
Warszawa

Wprowadzenie do maszynowego postrzegania

Od około 15 lat w literaturze informatycznej wymienia się nową problematykę, tzw. maszynowe postrzeganie (MP) (ang. *machine vision*). Formułuje się cel wyposażenia maszyn w takie możliwości, jakie ma ludzki system wzrokowy, aby mogły one postrzegać otoczenie w ich polu widzenia, rozumieć to, co spostrzegają, oraz podejmować stosowne, zaprogramowane akcje. MP jest zagadnieniem dziedziny sztucznej inteligencji (AI), dlatego budzi szczególne zainteresowanie. Widzenie jest bowiem związane nie tylko z oświetleniem i geometrią sceny lub odbijaniem światła, ale także (a może przede wszystkim) z intelektualnymi czynnikami rozpoznawania i rozumienia otoczenia.

W pierwszych latach badań w dziedzinie MP danymi wejściowymi były cyfrowe jednokolorowe obrazy światło-intensywnościowe. **Cyfrowy obraz światło-intensywnościowy** (ang. *light-intensity image*) jest prostokątną macierzą wartości oznaczających jaskrawość obrazu w punktach regularnej siatki i nie zawiera, w jawnej postaci, informacji o głębi sceny. Ludzie potrafią szybko i łatwo poprawnie wnioskować o elementach głębi w obrazie, lecz zapewnienie wnioskowania automatycznego o takich zależnościach jest nadzwyczaj trudne.

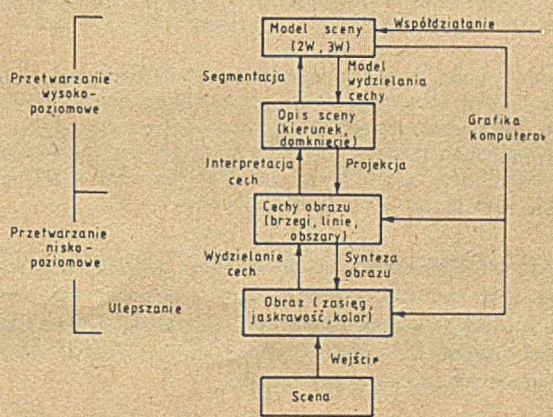
Obecnie większość badań MP dotyczy postrzegania trójwymiarowego (3W). Dwa najważniejsze problemy to: rozpoznanie obiektów trójwymiarowych na podstawie pojedynczych dwuwymiarowych widoków z dowolnego położenia oraz interpretacja obrazów dwuwymiarowych jako scen trójwymiarowych.

Jest w zasadzie niemożliwe zobaczenie zasłoniętej części obiektu. Jeżeli jednak wiadomo coś a priori o scenie lub jeśli są dane modele, tzn. wzorce obiektów mogących wystąpić na scenie, to można wydedukować również obraz zasłoniętych części i rozpoznać obiekt. Ludzie mają, na przykład, zdolność domyślania się wypukłości bądź wklęsłości brzegów z fotografii przedmiotów.

W ostatnich latach MP zrobiła zauważalny postęp: stały się dostępne różnego rodzaju czujniki obrazowe do przestrzennego pomiaru sceny. Z czujników tych uzyskuje się dane zasięgowe o scenie, tzw. **mapę głębi sceny** (ang. *scene depth map*). Jest to tablica odległości między płaszczyzną soczewki czujnika a powierzchniami obiektów sceny w polu widzenia czujnika (branych wzdłuż promieni wychodzących z węzłów pewnej regularnej siatki). Otrzymuje się wówczas zależności między obszarami mapy głębi sceny a trójwymiarowymi kształtami obiektów sceny. Ponieważ informacja o zasięgu zależy jedynie od geometrii, a nie zależy od oświetlenia i odbić, więc nie ma problemów związanych z wyznaczaniem powierzchni i z cieniami. Dlatego rozpoznawanie kształtów na podstawie mapy głębi jest na ogół łatwiejsze. Dane zasięgowe dostarczają również bardziej wiarygodnych informacji o geometrii sceny niż intensywność światła, ułatwiając przez to rozwiązanie problemów o większej złożoności, jak np. dopasowywanie trójwymiarowych wzorców przesłaniających się obiektów.

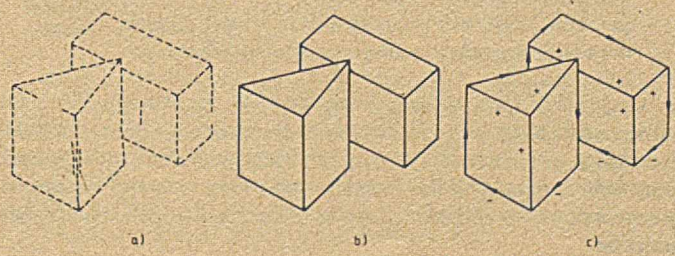
Metody rozumienia scen trójwymiarowych nie są jednak jeszcze ustabilizowane. Również podstawowe techniki trójwymiarowego MP nie są jeszcze ustalone, a większość praktycznych systemów rozpoznających dotyczy obiektów dwuwymiarowych.

Procesy postrzegania trójwymiarowego mają oczywiście wiele składowych, zawierających wykonywanie określonych operacji. Sekwencja operacji przetwarzania zależy od zadań, obiektów i wstępnej informacji. Typowy proces dla światło-intensywnościowego obrazu prostej sceny, z klockiem i trójkątnym graniastosłupem, wygląda następująco (rys. 1). Z wejściowego obrazu są wydzielane brzegi jako punkty nagłej zmiany intensywności światła (rys. 2a). Ponieważ brzegi te są cechami obrazu, nie muszą odpowiadać rzeczywistym brzegom powierzchni obiektów. Po połączeniu brzegów otrzymuje się ciągi brzegów, czyli rysunek liniowy (rys. 2b), będący w dalszym ciągu cechą obrazu. Ten etap przetwarzania, tj. przetwarzanie do stadium otrzymania cech obrazu, nazywa się **przetwarzaniem niskopoziomowym** (ang. *low-level processing*).



Rys. 1. Główne składowe i typy informacji występujące w MP

W drugim etapie przetwarzania, zwanym **przetwarzaniem wysokopoziomowym** (ang. *high-level processing*) otrzymany w pierwszym etapie rysunek podlega interpretacji jako scena trójwymiarowa, przez przypisanie fizycznych znaczeń liniom za pomocą etykiet. Zinterpretowany rysunek liniowy jest opisem sceny. Na rysunku linie z etykietami reprezentują cechy (brzegi powierzchni) sceny. Teraz rysunek jest dzielony na części (rys. 2c). Każda część jest rozpoznawana za pomocą dopasowywania do wzorców obiektów, wprowadzonych wcześniej do systemu postrzegania.



Rys. 2. Przykład procesu rozumienia obrazu: a) wydzielanie brzegów z obrazu wejściowego, b) rysunek liniowy, c) zinterpretowany rysunek liniowy

Przedstawiony przykład jest nazywany **procesem dół–góra** (ang. bottom-up process) ponieważ rozpoczyna się od operacji podstawowych na obrazie, a kończy na dopasowywaniu obiektu do wzorca (na rys. 1 od dołu do góry).

Proces dół–góra nie zawsze jest efektywny. Jeżeli na klasę obiektów i ich orientację są nałożone dostateczne ograniczenia, obraz może być

wprost dopasowywany do wzorców obiektów (oczekiwanych obrazów obiektów). Bardziej wydajny jest **proces góra–dół** (ang. *top–down process*) ponieważ można otrzymać wynik przy mniejszej liczbie obliczeń. Z tych powodów często wykorzystuje się go w zastosowaniach przemysłowych, takich jak rozmieszczenie układów o wielkim stopniu scalenia.

Proces góra–dół jest niepraktyczny, jeśli narzucono liczne ograniczenia. Wówczas jest wymagana duża liczba wzorców, a dopasowywanie musi być wykonane na wysokim poziomie. Jednakże, aby zrozumieć złożoną scenę zarówno proces góra–dół jak i proces dół–góra są nieodpowiednie. Wydaje się, że rozpoznawanie złożonej sceny wymaga dopasowywania do wzorców na pewnym pośrednim poziomie; wzorce te mogą być wprowadzone do systemu lub przekształcone automatycznie z poziomu zadanego. Rozpoznawanie np. klocków o różnych rozmiarach można osiągnąć za pomocą dopasowywania na poziomie cech obrazu. W ogólności wynik procesu dół–góra może być użyty w wysokopoziomowym przetwarzaniu, a wynik przetwarzania wysokopoziomowego może być wykorzystany w przetwarzaniu niskopoziomowym.

Człowiek spotykając obiekt, którego nigdy przedtem nie widział, jest zwykle w stanie zgromadzić informację o tym obiekcie, oglądając go z kilku różnych stron. Proces ten, tzn. gromadzenie informacji o obiekcie, jest nazywany **formowaniem wzorca** (ang. *pattern forming*). Natomiast znane sobie obiekty człowiek może identyfikować (w zasadzie z dowolnego punktu obserwacji) już bez dodatkowych zabiegów; co więcej, jest on zdolny również identyfikować obiekty i określać ich usytuowanie na jednokolorowych zdjęciach.

Ponieważ chodzi o proces rozpoznawania automatycznego (komputerowego), więc dane wejściowe z czujników muszą mieć postać cyfrową. Termin **scyfrowane dane czujnikowe** (ang. *digitized sensor data*) oznacza wejściową macierz numerycznych wartości (reprezentujących jaskrawość obrazu, głębię sceny lub inny parametr skalarny) oraz informację, jak te wartości zostały uzyskane.

Dla potrzeb systemów wizyjnych szerokiego zastosowania, takich jak automatyczna kontrola, zbieranie danych o scenie czy nawigacja automatycznych pojazdów, definiuje się **1–dowolnowidokowe algorytmiczne rozpoznawanie** (ang. *single-arbitrary-view algorithmic recognition*) trójwymiarowych obiektów następująco:

1. Dany jest zestaw oznakowanych bryłowych obiektów, przy czym
a) każdy z tych obiektów może być badany w dowolny sposób, byle tylko nie spowodowało to jego zmiany,
b) mogą być tworzone wzorce obiektów na podstawie informacji pochodzących z badania tych obiektów;

2. Zadane są: scyfrowane dane wejściowe, odpowiadające pewnemu polu widzenia sceny, informacje uzyskane podczas formowania wzorców oraz lista rozróżnialnych obiektów sceny, należy zbadać:
a) czy w danych wejściowych pojawia się obiekt?
b) jeśli tak, to w jakim czasie (tzn. jak długo on występuje)?
c) dla każdego pojawienia się obiektu należy określić:
- położenie w danych wejściowych,
- położenie względem znanego układu współrzędnych,
- orientację względem znanego układu współrzędnych;

3. Opcjonalnie, jeśli w danych wejściowych są obszary, które nie odpowiadają żadnemu z obiektów na liście, system powinien scharakteryzować je w taki sposób, aby stały się rozpoznawalne w późniejszych obrazach. Przypuszczalnie występuje tam obiekt nieznany systemowi, a system powinien próbować dowiedzieć się czegokolwiek o nieznanym obiekcie.

Punkt 3. jest zadaniem częściowej budowy wzorca na podstawie danych, które nie mogą być wytłumaczone w terminach już istniejących wzorców. Nie jest to rozpoznawanie, ponieważ coś w danych wejściowych jest systemowi całkowicie nieznane; system jest zmuszony do uczenia się przez doświadczenie.

Możliwości systemów rozpoznających powinny, oczywiście, być mierzalne, np. liczbą błędów popełnianych podczas wykonywania zadań z wybranego problemu na standaryzowanych zbiorach danych wejściowych. Porównywanie systemów rozpoznających można wówczas wykonywać ilościowo za pomocą wskaźnika możliwości systemu, który zestawia typy i wielkości różnych błędów. Niestety, informacja w literaturze na temat istniejących systemów nie pozwala praktycznie na ich ilościowe porównania, ponieważ różni badacze nie oceniają swoich systemów w jednolity sposób.

Główne kierunki badań maszynowego postrzegania

We wczesnym stadium rozwoju MP było w świecie kilka zaledwie zespołów badawczych zajmujących się tą tematyką. Obecnie tematyka ta jest rozwijana na uniwersytetach, w instytutach i firmach wielu krajów. Początkowo prace w MP ograniczały się do przetwarzania obrazów i rozpoznawania form. Typowym przykładem tych prac było rozpoznawanie znaków i analiza chromosomów. Większość z nich dotyczyła klasyfikacji dwuwymiarowych form wyizolowanych (lub podzielonych) obiektów.

Uważa się, że prace w trójwymiarowym MP zapoczątkował Roberts we wczesnych latach sześćdziesiątych, w Lincoln Laboratory (MIT) [5, 6, 7]. Jego program „rozumiał" sfotografowane sceny z wielościanem. Idea działania tego programu była następująca: na wejściowym obrazie punkty brzegowe były wykrywane i łączone w ciągi. Następnie znajdowano równania tych ciągów, a otrzymany rysunek liniowy był opisem obrazu. Do rozpoznania każdego wielościanu program Robertsa zawierał trójwymiarowe wzorce. Reprezentacja wzorca była jak w grafice komputerowej: wielokąt był reprezentowany przez współrzędne wierzchołków. Wierzchołki te były z rysunku wybierane i dopasowywane do kolejnych wzorców. Gdy pasowanie było pomyślne, obiekt był całkowicie rozpoznany (rozpoznane były również trójwymiarowe struk-

Konferencje

W dniach od 21 do 24 maja 1990 r. odbyła się w Kazimierzu Dolnym (w Domu Prasy) **I Krajowa Konferencja Naukowa pn. „Grafika Komputerowa i Przetwarzanie Obrazów" (GKPO'90).**

Jej celem było spotkanie i zapoznanie się oraz nawiązanie ściślejszej współpracy osób zajmujących się problematyką komputerowego postrzegania, przetwarzania i generowania informacji obrazowej.

Na Konferencję nadesłano 53 referaty (w tym trzy z Kanady), których zdecydowana większość została zakwalifikowana jako mieszczące się w zakresie problemowym Konferencji i mające odpowiedni poziom merytoryczny. Do wygłoszenia wybrano 41 referatów. Zostały one podzielone (umownie) na sześć grup tematycznych i były wygłoszone w czasie oddzielnych sesji, przy czym każda z nich rozpoczynała się (dobranym tematycznie) wykładem osoby zaproszonej, integrującym, w pewnym sensie, tematykę referatów nadesłanych i zgrupowanych w danej sesji.

A oto tematy sesji:

1. Problemy przetwarzania i obrazowania. Wygłoszono siedem referatów. Wykład pt. „Możliwości rozwojowe komputerowego przetwarzania obrazów naturalnych" wygłosił prof. dr hab. inż. Juliusz Kulikowski, z Instytutu Biocybernetyki i Inżynierii Biomedycznej PAN.

2. Problematyka sprzętowa w symulatorach lotu. Wygłoszono osiem referatów. Wykład pt. „Wizualizacja scen dla potrzeb symulatorów lotu" wygłosił prof. dr hab. inż. Jan Zabrodzki, z Instytutu Informatyki Politechniki Warszawskiej.

3. Złożoność obliczeniowa algorytmów i zastosowania. Wygłoszono sześć referatów. Wykład pt. „Złożoność obliczeniowa problemów grafiki komputerowej" wygłosił dr inż. Władysław Skarbek, z Instytutu Podstaw Informatyki PAN.

tury jego zasłoniętej części). Różniło się to zdecydowanie od przetwarzania obrazów dwuwymiarowych.

Roberts zajmował się wykrywaniem i łączeniem brzegów, rysunkami liniowymi, trójwymiarowymi wzorcami oraz dopasowywaniem. Jednak jego program był właściwie ciągiem kombinacji tych procesów od niskiego do wysokiego poziomu przetwarzania. Takie podejście (dół–góra) miało ograniczenia, ponieważ na etapie wykonywania programu można było używać wyników uzyskanych jedynie w poprzednim etapie. Dlatego też obrazy wejściowe musiały być proste i dostatecznie czyste (nie zaszumione) na każdym etapie, aby proces przebiegał poprawnie.

W późnych latach sześćdziesiątych podjęto prace również w kilku innych ośrodkach badawczych: Stanford University, Stanford Research Institute (SRI), Institute of Fundamental Research TATA, Bombaj, (IFR TATA) [6, 7] oraz Edinburgh University. Powstające tam konstrukcje robotów można było nazwać robotami pierwszej generacji. Zadaniem systemu wizyjnego takiego robota było rozpoznanie klocków na zadawanym (kontrolowanym) tle. Głównym zamysłem tych prac było przetestowanie wykonalności zadań MP i zademonstrowanie możliwości zintegrowanych robotów. Eksperymenty te pokazały, że rzeczywiste dane wzrokowe różnią się od oczekiwanych do tego stopnia, że ciągi wykrytych punktów brzegowych obejmują nie istniejące brzegi, a nie zawierają istniejących.

W 1968 roku prace projektowe nad inteligentnymi robotami podjęto również w Japonii – w Electrotechnical Laboratory (ETL) i w firmie Hitachi Company. Wyniki zademonstrowano w 1970 r. Robot firmy Hitachi budował zestawy z klocków, ETL natomiast pokazała wiele podstawowych technik, łącznie z zadaniem zbierania klocków, w którym wykorzystano wizyjne sprzężenie zwrotne.

Chociaż pierwsza generacja inteligentnych robotów miała ograniczone możliwości, wywarły one wielki wpływ na prace późniejsze. Podczas realizacji tych projektów zrodziło się i rozwijało wiele pomysłów i tematów badawczych oraz dostrzeżono wiele problemów, wymagających takich rozwiązań, aby można było usunąć istniejące ograniczenia robotów.

W końcu lat sześćdziesiątych Guzman (MIT) jako pierwszy rozpoczyna badania nad interpretacją rysunków liniowych, otrzymywanych na podstawie analizy zdjęć wielościanów [6, 7]. W przeciwieństwie do pracy Robertsa jego program SEE starał się podzielić rysunek między bryły bez użycia wzorców obiektu. Guzman przypisywał znaczenie kształtom połączeń, które pojawiają się na rysunkach liniowych i wprowadzał wiele heurystycznych reguł do grupowania obszarów na rysunkach liniowych. Chociaż program SEE poprawnie postępował przy wielu złożonych rysunkach, były jednak niepowodzenia w przypadku innych. Wytłumaczenie pomyłek było trudne, ponieważ decyzja opierała się na heurystyce.

Również w końcu lat sześćdziesiątych Reddy i Narasimhan, z IFR TATA, do obrazów trójwymiarowych scen złożonych z klocków o różnych kształtach i rozmiarach, zastosowali z powodzeniem metody gramatycznego rozbioru obrazu w terminach cech oraz relacji w rodzaju: *z boku, z przodu, wyżej, niżej, wewnątrz, na zewnątrz, razem, oddzielnie, liniopodobne, plamopodobne* itp., wykorzystując do tego języki COMPAX i PAX II [6].

W 1971 r. Huffman w University of California, w Santa Cruz, oraz Clowes w Sussex University niezależnie zaproponowali teorię rysunków liniowych. Linie w rysunku mogą odpowiadać wypukłym, wklęsłym lub nieciągłym (zasłoniętym) brzegom w trójwymiarowej scenie. Interpreta-

Grafika Komputerowa i Przetwarzanie Obrazów

4. Reprezentacje obiektów graficznych. Wygłoszono sześć referatów. Wykład pt. „Metody reprezentacji (opisu) obiektów graficznych" wygłosił dr Michał Jankowski, z Instytutu Informatyki Uniwersytetu Warszawskiego.

5. Maszynowe postrzeganie. Wygłoszono siedem referatów. Wykład pt. „Maszynowe postrzeganie. Próba formalizacji" wygłosił doc. dr hab. inż. Wojciech Mokrzycki, z Instytutu Podstaw Informatyki PAN.

6. Modelowanie geometryczne. Wygłoszono pięć referatów. Wykład pt. „Modelowanie geometryczne w systemach CAD-CAM" wygłosił doc. dr hab. inż. Krzysztof Marciniak, z Instytutu Automatyki Przemysłowej Politechniki Warszawskiej.

Odbyła się również sesja specjalna poświęcona grafice komputerowej w sztuce, na której przeprowadził wykład (połączony z pokazem przezroczy) pt. „Zarys historii sztuki komputerowej" doc. Jan Pamuła, z Akademii Sztuk Pięknych w Krakowie. Wykład ten, mimo że odbywał się w porze wieczornej, wzbudził bardzo duże zainteresowanie i z tego właśnie powodu trwał dwukrotnie dłużej niż planowano. W tej samej sesji wystąpił również z wykładem i pokazem udźwiękowionych animacji komputerowych doc. Wojciech Bruszewski, z PWSSP w Poznaniu. Jego wystąpienie wzbudziło bodajże największe zainteresowanie uczestników Konferencji, trwało ponad dwukrotnie dłużej niż planowano i, jako jedyne wystąpienie na Konferencji, zakończyło się (po godzinie 23^{00}), oklaskami dla referenta.

Sesje wieczorne (oprócz pierwszej, poświęconej sztuce komputerowej) były zarezerwowane dla demonstracji systemów użytkowych. Odbyło się dziewięć pokazów opracowanych systemów komputerowych w zakresie grafiki, przetwarzania obrazów, robotyki bądź CAD-CAM.

Przez cały czas trwania Konferencji odbywała się wystawa opracowań własnych uczestników. Urozmaiceniem Konferencji był koncert organowy w kościele farnym oraz zwiedzanie zabytków Kazimierza.

W Konferencji wzięło udział ok. 70 osób z 28 jednostek organizacyjnych z całego kraju. Najliczniej reprezentowany był IPI PAN (organizator Konferencji) – 10 przyjętych referatów (14 uczestników), a dalej w kolejności: Instytut Informatyki PW – 9 (7), Instytut Automatyki Przemysłowej PW – 5 (5), Instytut Informatyki UW – 3 (2), Instytut Informatyki UJ – 2 (4), Katedra Informatyki i Automatyki Morskiej Politechniki Szczecińskiej – 2 (1) i z innych.

Organizator GKPO' ma zamiar organizować międzynarodowe konferencje co dwa lata pod nieco zmienionym hasłem „Grafika i Widzenie". Jakiekolwiek sugestie na ten temat, jak również chęć uczestniczenia w konferencji najbliższej w 1992 r. prosimy kierować pod adresem Instytutu Podstaw Informatyki PAN, 00-625 Warszawa, ul. Polna 18/20, Konferencja „Grafika i Widzenie", lub do Redakcji INFORMATYKI.

Konferencja GKPO'90 zaktywizowała w znacznym stopniu środowisko informatyków zajmujących się zagadnieniami komputerowego przetwarzania informacji obrazowej. Niejako przy okazji Konferencji powstało wiele opracowań dotyczących tej tematyki, stąd zamysł wydania niniejszego specjalnego numeru INFORMATYKI pod hasłem: „Grafika i Widzenie", poświęconego tym problemom. Prosimy Czytelników o zdanie na temat monotematycznych numerów INFORMATYKI.

W.M.

cję rysunków liniowych zredukowano do oznaczenia tego typu linii rysunku wśród wszystkich linii. Dla danej klasy obiektów klasyfikuje się kształty połączeń oraz wylicza typy linii wymaganych w każdym połączeniu. Interpretacja polega natomiast na określeniu typów linii, które są zgodne z dozwolonymi kształtami połączeń.

W 1972 r. Waltz w MIT rozszerzył tę teorię na większy zbiór typów linii, łącznie z brzegami cieni. Jednak ta metoda obejmowała jedynie poprawne rysunki. Zasady interpretacji rysunku liniowego rozszerzono w kilku różnych kierunkach: interpretacji niedokończonych rysunków liniowych obiektów krzywoliniowych oraz określania realizowalności rysunków liniowych.

W 1972 r. Falk ze Standford University opracował program INTERPRET do interpretowania rysunków liniowych z brakującymi liniami. Program INTERPRET kawałkował linie oraz wystawiał je zgodnie z heurystycznymi regułami i trójwymiarowym wzorcem wielościanu. Poprawność interpretacji była sprawdzana przez dopasowywanie zmodyfikowanego rysunku do oryginalnego.

W 1970 r. Brice i Fennema z SRI zaproponowali tzw. metodę obszaru. Pozwalała ona wykrywać jednorodne obszary przez wchłanianie podobnych mniejszych. Popplestone i inni z Edinburgh University uogólnili tę metodę i zastosowali ją do rozpoznawania obiektów krzywoliniowych. W latach późniejszych opracowano metody podziału obszaru oraz metody podziału i łączenia obszarów. Metody obszaru stosuje się w większości projektów analizy obrazów kolorowych.

W 1971 r. Shirai (MIT) zaprezentował wyniki studiów dotyczących rozpoznawania wielościanów z wykorzystaniem danych zasięgowych. W 1973 r. Agin i Rinford ze Stanford University opracowali reprezentację danych zasięgowych z krzywoliniowych obiektów, z wykorzystaniem uogólnionych cylindrów. Otrzymywanie typów brzegów lub równań powierzchni trójwymiarowych bezpośrednio z danych zasięgowych jest łatwe. W latach siedemdziesiątych w ETL, korzystając z tych udogodnień, opracowano metodę opisu scen z wielościanów i nieskomplikowanych krzywoliniowych obiektów, w terminach brzegów prowadzonych za pomocą słownika łączącego, oraz metodę do rozpoznawania obiektów zasłaniających się.

Opracowanie metody interpretowania rysunków liniowych wywołało pytanie o ich realizowalność. Nie było wiadomo, czy interpretowane rysunki liniowe rzeczywiście istnieją (tj. czy określony zestaw linii może być rysunkiem liniowym jakiegoś obiektu geometrycznego). Do sprawdzenia ich realizowalności zaproponowano **dwoistą przestrzeń** (ang. *double space*) nazywaną również **przestrzenią gradientową** (ang. *gradient space*), (reprezentuje kierunek normalnych do powierzchni). Za pomocą transformacji rysunku liniowego do przestrzeni gradientowej wykryto kilka „niemożliwych" rysunków liniowych. Jednak zaakceptowanie wyniku testu za pomocą tej metody nie jest wystarczającym warunkiem realizowalności rysunku liniowego. Falk zaproponował, aby do określania trójwymiarowej geometrii wielościanów użyć równań płaszczyzn [2]. Metodę tę rozszerzono następnie do teorii algebraicznej i stosowano do określenia realizowalności dowolnych zinterpretowanych rysunków liniowych. Chociaż przestrzeń gradientowa nie jest odpowiednia do całościowego rozwiązania interpretacji rysunków liniowych, jednak wzbogaca trójwymiarowe MP o pożyteczną reprezentację obiektów. Na przykład, rekonstrukcję powierzchni trójwymiarowej wygodnie jest wytłumaczyć w terminach przestrzeni gradientowej.

Różne osoby zwykle tak samo interpretują obraz lub nawet niepoprawny rysunek liniowy wydzielony ze sceny, ponieważ mają wiedzę o obiektach na scenie. Minsky i Papert zaproponowali, jako sposób interpretacji obrazu, podejście wielohierarchiczne, zamiast podejścia góra–dół. W 1972 r. Shirai zrealizował w MIT wielohierarchiczny program do poszukiwań semantycznych. Program, na podstawie zależności w wielościanie oraz linii otrzymanych poprzednio, „domyślał się", gdzie linia powinna być znaleziona i jaki to może być typ linii.

W 1973 r. Yakimovsky i Feldman, ze Stanford University, rozwinęli semantyczny analizator obszaru, który rozpoczynając z niekompletnych wyników metody obszaru, otrzymywał „lepsze obszary" z interpretacjami opartymi na statystycznej teorii decyzji. W 1976 r. zespół z SRI zaproponował metodę segmentacji obszaru z tzw. **wiedzioną interpretacją**. Również w tej metodzie były wchłaniane „niedoskonałe obszary", otrzymywane metodą obszaru na podstawie modelu sceny, który reprezentował geometryczne zależności między obszarami.

W końcu lat sześćdziesiątych Horn z MIT zaproponował w swojej rozprawie doktorskiej metodę odzyskiwania kształtów z cienia (ang. *shape from shading*). W metodzie tej wykorzystano zależność między intensywnością światła obrazu a normalnymi do powierzchni trójwymiarowej bryły. Kształt uzyskuje się w wyniku rozwiązania równań różniczkowych wyprowadzonych z tych zależności. Jeżeli odbijalność powierzchni jest znana, to intensywność światła w punkcie na obrazie ogranicza położenie normalnej do powierzchni w przestrzeni gradientowej do jednowymiarowego. W połowie lat siedemdziesiątych Horn opracował metodę stereo-fotometrycznego uzyskiwania normalnych do powierzchni z wielokrotnych obrazów, oświetlanych z różnych kierunków.

W 1978 r. Barrow i Tenenbaum z SRI analizowali problem uzyskiwania charakterystyk sceny z obrazu. Charakterystyki sceny, które były rozróżnianie z pewnego obrazu, nazwali **wewnętrznymi charakterystykami** (ang. *internal characteristics*). Dowiedli oni, że wewnętrzne charakterystyki są osiągalne za pomocą pewnego procesu bez semantyki, wykorzystującego ograniczenia wyprowadzone z założeń dotyczących natury sceny i fizyki procesu obrazowania. Do odzyskiwania kształtów wykorzystywano również ograniczenia występujące w geometrii trójwymiarowej.

W końcu lat siedemdziesiątych Kender, w Carnegie-Mellon University (CMU), uzyskał normalną płaską powierzchnię z regularnie wyrównanych tekstur, a Kanade zaproponował metodę odzyskiwania kształtów, oparą na założeniach symetrii.

Wielki wpływ na myślenie o MP miały prace Marra o teorii ludzkiego widzenia, prowadzone w końcu lat siedemdziesiątych. Dowiódł on, że proces widzenia przekształca orginalne obrazy na trzy poziomy reprezentacji: szkic pierwotny, szkic dwu i półwymiarowy ($2^1/_2$ W) oraz trójwymiarową reprezentację. Pierwotny szkic jest kolekcją cech obrazu: różnych zmian jasności światła. W szkicu dwu i półwymiarowym pamięta się informację o płaszczyznach. Trójwymiarowa reprezentacja każdego obiektu jest natomiast wyrażeniem obiektu w jego współrzędnych centralnych. Zespół Marra z MIT zbudował model tej teorii w postaci programów komputerowych. Obserwacje systemów ludzkiego widzenia oraz psychologia doświadczalna były zamodelowane w wydzielaniu cech, skupieniu cech (typu łączenia brzegów) oraz wyobraźni przestrzennej. Idee te pociągnęły za sobą wiele projektów badawczych, m.in. badanie przestrzenności obiektu na podstawie cech obrazu.

Dziedziny pokrewne

Jest wiele dziedzin, w których korzysta się z technik przetwarzania informacji wzrokowej, np. przetwarzanie obrazów, klasyfikowanie i rozpoznawanie wzorców, rozumienie obrazów lub grafika komputerowa. Uważa się, że rozumienie obrazów ma podobne znaczenie jak MP. Pozostałe dziedziny, w szczególności przetwarzanie obrazów i rozpoznawanie obrazów, dotyczą całkowicie podobnych zagadnień.

Celem przetwarzania obrazów (zdjęć) jest, w wąskim sensie, ulepszenie wejściowego obrazu dla potrzeb obserwatora. Na przykład, rozogniskowane obrazy są uwyraźniane lub jest zmieniany kontrast, przez co mogą być wzmocnione poszczególne elementy obrazu. Szybkie przetwarzanie obrazu można osiągnąć za pomocą zarówno procesorów cyfrowych, jak i analogowych lub nawet sprzętu optycznego. Ponieważ przetwarzanie obrazów jest wykonywane dla potrzeb jego oceny przez człowieka, szczegóły (lub parametry) przetwarzania są również formułowane przez operatora. Dlatego system przetwarzający nie obejmuje znaczenia obrazu.

Celem klasyfikowania wzorców jest zakwalifikowanie określonego obiektu do wcześniej wskazanej klasy. Typowym przykładem jest rozpoznawanie ciągu znaków. Każdy wejściowy znak jest wydzielany

z ciągu oraz jest zapamiętywana jego klasa. Klasyfikowanie wzorców można rozważać jako specjalny przypadek MP, w którym na wyjściu jest tylko nazwa obiektu.

Wzorce są klasyfikowane zwykle na podstawie zbioru cech wydzielonych z wejściowego obrazu. Zbiór cech jest porównywany z cechami przewidywanymi dla każdej klasy wzorców, a decyzja jest podejmowana na podstawie kilku kryteriów. Ponieważ celem jest wyróżnienie jednej z klas, system klasyfikowania nie musi dotyczyć treści obrazu. Najlepszy zbiór cech można wybrać korzystając z teorii statystyki. W prostym przypadku mogą być używane cechy łatwe do wydzielenia. Jeżeli, na przykład, zadaną klasą znaków jest pięć pierwszych znaków alfabetu łacińskiego, to zbiorem cech użytych do klasyfikowania może być liczba dziur i liczba punktów końcowych na znakach. Dlatego wybrany zbiór cech nie musi opisywać obrazu. Decydowanie jest proste, gdy został wydzielony zbiór cech, pokrywający się z jedną z klas wzorców. Jednak w praktyce zbiór wydzielonych cech nie jest identyczny z żadnym z wzorców; wówczas należy podjąć decyzję, który z wzorców jest najbardziej podobny. W takiej sytuacji często korzysta się ze statystycznej teorii decyzji.

W okresie wcześniejszym klasyfikowanie wzorców było jedynie dziedziną badań rozpoznawania wzorców. Z tych historycznych powodów termin rozpoznawanie wzorców (ang. *pattern recognition*) jest, w wąskim sensie, odpowiednikiem klasyfikowania wzorców.

Grafika komputerowa (GK) natomiast dotyczy zagadnień prawie przeciwstawnych do zagadnień MP. Jej zadaniem jest synteza obrazu sceny na podstawie jej opisu matematycznego w terminach modelu lub pewnych zaobserwowanych danych. Jest to zupełnie odwrotne zadanie niż w MP, gdzie celem jest stworzenie opisu sceny na podstawie jej obrazu. Jednak obie te dziedziny dotyczą tej samej fizycznej rzeczywistości. Dlatego podstawowe koncepcje i problemy dotyczące tych dwóch dziedzin są takie same, chociaż mają inny aspekt. MP i GK jak gdyby spinają klamrami całą problematykę przetwarzania i rozumienia informacji obrazowej. Problemami, które mogą pojawić się w GK są tzw. problemy kierunkowe; może – na przykład – wystąpić potrzeba obliczenia widocznych powierzchni i linii dla zasłaniających się obiektów lub intensywności oświetlenia na podstawie modelu geometrycznego sceny i modelu odbić światła od jej obiektów. W GK najważniejsze są prostota algorytmów i wydajność pamiętania, a jej rozwój łączy się ściśle z rozwojem sprzętu i sprawami współdziałania operatora z systemem. Chociaż danymi wejściowymi w GK jest zwykle opis obrazu, jednak obrazy rzeczywiste również mogą być danymi wejściowymi. Są one wykorzystywane do generowania nowych obrazów po dokonaniu na nich (przez użytkownika) wymaganych zmian.

Techniki GK w zastosowaniu do potrzeb MP wykorzystuje się do przekształceń wzorców wysokopoziomowych we wzorce niskopoziomowe, jak na rys. 1.

Ostatnio również obrazy realistyczne mogą być generowane z trójwymiarowych opisów sceny za pomocą tzw. **fraktali.** Zagadnienie reprezentowania trójwymiarowego opisu jest również ważne w trójwymiarowym MP. Wspólnym problemem wymagającym wspólnych rozwiązań jest rozwój metod opisu ruchomych obiektów i wiele innych zagadnień.

Matematyczne sformułowanie problemu

Matematyczne sformułowanie problemu chroni przed możliwymi dwuznacznościami. W terminach matematycznych rozpoznawanie obiektów, zarówno mapo-głębiowe, jak i światło-intensywnościowe, traktuje się jako **uogólnione inwersyjne odwzorowanie zbioru** (ang. *generalized inverse set mapping*).

Niech scena zawiera N_c obiektów O_i, $N_c = \{O_i\}$. Liczba obiektów rozróżnialnych niech będzie N_{or}; a więc $N_c \geqslant N_{or}$ (dwa obiekty są nierozróżnialne, jeśli rozpoznający nie może powiedzieć o nich odmiennie w terminach kształtowych). Niech O_i oznacza i-ty obiekt rozróżnialny, a N_i – liczbę pojawień się tego obiektu. W ogólności

$$N_c = \sum_{i=1}^{N_{or}} N_i$$

Ludzie mogą rozpoznawać bardzo dużą, w zasadzie nieograniczoną, liczbę obiektów, w zależności od osobistego doświadczenia. Liczba obiektów, które mogą być rozpoznane przez system rozpoznający, zależy od zastosowania i „wytrenowania" systemu. Aby uniknąć wątpliwości, co jest obiektem a co zbiorem obiektów, każdy obiekt powinien mieć własny układ współrzędnych oraz listę podobiektów.

Na początek będą rozważane jedynie proste obiekty bez składowych i tylko z jednym wystąpieniem. Będzie wówczas $N_{or} = N_c$. Ogólny przypadek wielokrotnych wystąpień obiektów ze składowymi nie jest koncepcyjnie odmienny, chociaż występują prolemy zapisu i są możliwe trudności wykonawcze na wyższych poziomach procesu rozpoznawania.

Niech będzie dany prostokątny układ współrzędnych obiektu mający początek w środku (masowym) obiektu. Niech istnieje również układ współrzędnych sceny. Obiekty są lokalizowane w przestrzeni względem układu sceny za pomocą parametrów przesunięcia t i obrotu θ. Liczba parametrów dla każdego z wektorów zależy od wymiaru problemu rozpoznawania; dwuwymiarowy problem wymaga trzech parametrów (dwa dla przesunięcia, jeden dla obrotu), trójwymiarowy problem – sześciu parametrów ($t = \langle t_x, t_y, t_z \rangle$, $\theta = \langle \theta_x, \theta_y, \theta_z \rangle$).

Model sceny S definiuje się jako zbiór uporządkowanych trójek $\langle$*obiekt, przesunięcie, obrót*$\rangle$:

$$S = \{\langle O_i, t_i, \theta_i \rangle\}_{i}^{N_{or}} = 0$$

Czujnik jest rozważany jako obiekt O_0 mający położenie t_0 orientację θ_0. Dla potrzeb 1-dowolno widokowego rozpoznawania interesujące są jedynie statyczne wartości parametrów (nie zmieniające się w czasie). Zbiór wszystkich przesunięć będzie oznaczany przez R^d, R – zbiór liczb rzeczywistych, a zbiór wszystkich obrotów przez R^r. Dla obiektów dwuwymiarowych $d = 2$, $r = 1$, a dla trójwymiarowych $d = 3$ i $r = 3$.

Zakładając, że czujnik wytwarza projekcję sceny w postaci mapy głębi sceny, można traktować tę projekcję jako matematyczny operator P (równoległy bądź perspektywiczny), który odwzorowuje elementy ze zbioru $\Omega = N_c \times R_d \times R^r$ w elementy zbioru F wszystkich skalarnych funkcji d-1 zmiennych, o wartościach rzeczywistych, nazywanych **funkcjami map głębi** (ang. *depth map functions*).

$$P: \Omega \rightarrow F$$

Projekcję zapisuje się jako:

$$f(x) = g_{O\,t\,\theta}(x) = P(O, t, \theta)$$

gdzie x jest wektorem (d-1 przestrzennych zmiennych) płaszczyzny soczewkowej czujnika. Parametry przestrzenne (obiektu) czujnika (położenie t_0 i orientacja θ_0) są niejawnie przyjętymi argumentami operatora rzutowania. Pozwala to uprościć wyrażenia, ponieważ w tym formalizmie jest tylko jeden czujnik. Niech funkcja mapy głębi będzie oznaczona przez f, gdy obiekt i jego parametry są nieznane, oraz przez g z indeksami dla przypadku znanego obiektu w znanym położeniu i orientacji. Zapis ten oznacza, że zbiór funkcji map sprzężonych z pojedynczym obiektem jest nieskończoną rodziną funkcji. Dwa z paramentrów obrotu w wektorze B mają szczególnie duży wpływ na tę rodzinę funkcji: trójwymiarowy kształt funkcji mapy głębi zmienia się, gdy obiekt obraca się. Parametry przesunięcia nie mają jakiegokolwiek wpływu na kształt (przy rzutowaniu prostopadłym i mają znikomy wpływ przy rzutowaniu perspektywicznym). Ponieważ obiekty nie zajmują całej przestrzeni sceny, więc jeżeli punkt *$(x, f(x))$ nie leży na powierzchni obiektu, to przypisuje się funkcji $f(x)$* wartość $-\infty$. Stąd projekcję zbioru M obiektów można zapisać jako:

$$f(x) = \max_{1 \leqslant i \leqslant M} g_{o_i t_i \theta_i}(x)$$

Problem rozpoznawania obiektów za pośrednictwem map głębi jest formułowany teraz następująco. Dana jest funkcja mapy głębi $f(x)$ wynikająca z projekcji trójwymiarowej sceny: trzeba określić zbiór możliwych obiektów (z odpowiednimi zbiorami parametrów przesunięć i obrotów), których rzuty tworzyłyby zadaną funkcję mapy głębi;

inaczej mówiąc trzeba określić zbiór wszystkich podzbiorów zbioru Ω taki, że:

$$\omega J = \{<O_j, t_j, \theta_j>\}_{j \in J}$$

rzutuje się w mapę głębi $f(x)$, gdzie J jest zbiorem wskaźników, które zależą od możliwych poprawnych interpretacji mapy głębi (rys. 3). Można to zapisać dokładniej, używając inwersyjnego odwzorowania. Dla każdej jedno obiektywowej funkcji mapy głębi istnieje odpowiednie inwersyjne odwzorowanie do wytworzenia wszystkich pojedynczych obiektów, które mogą tworzyć zadaną mapę głębi obiektu. Niech to zwrotne odwzorowanie P będzie oznaczone przez P^{-1}, gdzie

$$P^{-1}(f(x)) = \{<O,t,\theta> \in \Omega \mid P(O, T, \theta) = f(x)\}$$

Odwzorowanie inwersyjne przekształca zbiory ze zbioru o mocy rzędu początkowego odwzorowania w zbiory ze zbioru o mocy dziedziny:

$$P^{-1}: 2^F \rightarrow 2^\Omega$$

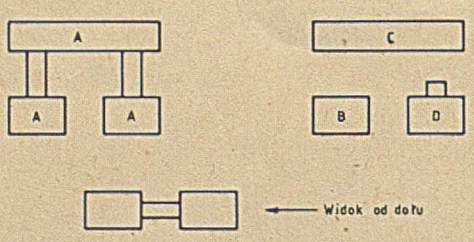
Dla naszych celów można ograniczyć wejściowe zbiory o mocy 2^F do pojedynczych funkcji map głębi; stąd 2^F zamienia się na F. Dla wieloobiektowych map głębi należy uogólnić zagadnienie do możliwych kombinacji tych obiektów. Stąd zadając $f(x) \in F$, szuka się uogólnionego odwzorowanie P^{-1} takiego, że

$$P^{-1}(f(x)) = \{\omega J \subseteq 2\Omega \mid \max_{j \in J} P(O_j, t_j, \theta_j) = F(x)\}.$$

Tak więc szukane odwzorowanie przekształca elementy przestrzeni funkcji mapy głębi w zbiór o mocy 2^Ω:

$$P^{-1}: F \rightarrow 2^{2^\Omega}$$

Zagadnienie mapo-głębiowego rozpoznawania obiektów może teraz być sformułowane w terminach zmodelowanej sceny następująco. Dany jest model sceny S z N_{or} pojedynczymi obiektami oraz pewna realizowana funkcja mapy głębi $f(x)$. Należy wyznaczyć takie odwzorowanie zbioru rzutowania zwrotnego $P^{-1}(f(x))$, aby otrzymać wszystkie możliwe wystąpienia (interpretacje) funkcji $f(x)$ w terminach modelu sceny.



Rys. 3. Przykład poprawnych interpretacji prostej sceny $P^{-1} = \{[A], [B, C, D], [D, C, B]\}$

W wielu wypadkach istnieje tylko jedna interpretacja sceny. Mimo tego, system wizyjny ogólnego przeznaczenia powinien generować listę wszystkich poprawnych interpretacji sceny, chociaż mogą wystąpić dwuznaczne sytuacje. Problem określenia następnego dobrego widoku, z którego dane czujnikowe powinny być pozyskane w celu wyeliminowania dwuznaczności, jest oddzielnym zagadnieniem. Ponieważ nie ma ogólnej teorii obliczenia tego odwzorowania, badacze mogą wybierać swoje własne metody. Proponuje się uogólnione odwzorowanie zwrotne, które najlepiej jest określić przez rozpoznanie oddzielnych obszarów powierzchniowych, sprzężonych z rodziną funkcji map głębi powierzchni $g_{O\,t\theta}(x)$ poszczególnych obiektów. Oznacza to podział zagadnienia rozpoznawania obiektów na podzagadnienia charakteryzowania powierzchni i zastosowanie algorytmów dopasowywania powierzchni, przy uwzględnieniu ograniczeń wynikających ze znanej struktury obiektów. Można rozszerzyć ten formalizm na problem rozpoznawania obiektów dla światło-intensywnościowych obrazów. Dodając do funkcji f mapy głębi operator I odbicia światła, otrzymuje się obraz intensywnościowy $i(x)$

$$i(x) = I(\max P(O_i, t_i, \theta_i))$$

Zagadnienie intensywnościowego rozpoznawania obiektów jest w ogólności wiele trudniejsze z powodu dodatkowej inwersji operatora I. Wymagana jest bowiem a priori wiedza o wszystkich powierzchniach odbijających światło i wszystkich źródłach światła na scenie (w dodatku do kształtów obiektu), aby było możliwe odwrócenie tego operatora I. Należy zauważyć, że najwięcej obliczeń wymaga wieloobiektowy operator domykania. Aby rozszerzyć model sceny w celu rozumienia intensywnościowych obrazów, niezbędne jest dodanie do pojedynczego obiektu wyczuwającego, który światło przyjmuje, obiektów generujących światło. Powstało wiele technik, np.: **kształt z cienia** (ang. *shape from shading*), **kształt z fotometrii stereo** (ang. *shape from photometric stereo*) i **kształt z tekstury** (ang. *shape from texture*) [7], będące jedynie próbą odwrócenia operatora I produkującego funkcję f mapy głębi. Interesująca jest obserwacja, że ludzie mają możliwość rozumienia obrazów nawet wówczas, gdy do cieniowania widzialnych powierzchni są używane operatory cieniujące inne niż operator oświetleniowo-odbijający. Na przykład, ludzie mogą poprawnie interpretować negatywy zdjęć lub pseudokolorowe obrazy, w których kolor lub zależności jasny-ciemny są kompletnie zniekształcone.

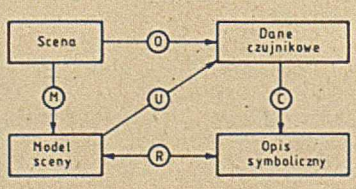
Składniki systemu rozpoznającego

Rozpoznanie implikuje znajomość czegoś, np. wzorca. Chociaż formowanie wzorca nie jest specjalnie wymagane w definiowaniu problemu MP, to jednak jest konieczne ze względu na okoliczności zagadnienia. Zakłada się więc, że dla potrzeb systemu wizyjnego jest **niezbędne formowanie wzorców obiektów**, mogących wystąpić w scenie, oraz że jest wymagany schemat reprezentacji do pamiętania danych o wzorcach tych obiektów. Model sceny jest więc niezbędnym składnikiem systemu rozpoznającego obiekty.

Do modelowania obiektów sceny do celów rozpoznawania stosuje się wiele różnych rodzajów wzorców, zarówno widokowo niezależnych, jak i widokowo zależnych. Preferuje się jednak widokowo niezależne reprezentacje obiektu, ponieważ **reprezentacja jest takim krytycznym czynnikiem w projektowaniu systemu rozpoznającego**.

Jeśli wiadomo jakiego obiektu wypatrywać w danych wejściowych, to trzeba jeszcze określić, jak będzie przebiegać samo rozpoznawanie. Do tego należy wybrać metodę dopasowywania danych wzorca do danych wejściowych. Najprostsze podejście dyktuje metodę przypadkowych poszukiwań, która pociąga za sobą transformowanie wszystkich możliwych kombinacji wszystkich znanych wzorców obiektów sceny we wszystkich możliwych rozróżnialnych orientacjach i wszystkich możliwych rozróżnialnych położeniach na formach danych wejściowych, a także wyliczanie wielkości błędu pasowania, który powinien być zminimalizowany. Uzyskanie minimum błędu pasowania odpowiada rozpoznaniu sceny. Takie podejście do problemu rozpoznawania może jednak pochłonąć ogromną ilość czasu przetwarzania nawet dla najprostszych scen. Niezbędny jest więc lepszy algorytm.

Ponieważ wzorce obiektu zawierają zwykle więcej informacji o obiektach niż dane wejściowe, nie można transformować danych wejściowych w pełne dane wzorca i dopasowywać ich do formatu danych wzorca. Pasowanie może jednak odbywać się z częściowymi danymi wzorca. Ta możliwość oraz naturalna potrzeba redukcji wielkiej liczby danych wejściowych powoduje, że zarówno z danych wejściowych, jak i wzorcowych jest wytwarzana pewna pośrednia dziedzina, nazywana **symboliczną dziedziną opisu sceny** (ang. *symbolic scene description domain*). Dane wejściowe są więc przetwarzane aż do osiągnięcia postaci symbolicznego opisu sceny, a dane wzorca są przekształcane w odpowiedni symboliczny opis sceny. Procedura pasowania może wówczas działać w pośredniej dziedzinie na wielkościach zwanych **cechami** (ang. *features*). Jednak najlepsze wyniki pasowania uzyskuje się wówczas, gdy konfiguracja wzorca dokładnie odpowiada scenie reprezentowanej przez dane wejściowe. Tak więc procedura pasowania i pośrednie symboliczne środki opisu sceny również powinny być niezbędnymi składnikami systemu rozpoznającego.



Rys. 4. Współdziałanie między składowymi systemu postrzegania

Współdziałanie między poszczególnymi składnikami systemu rozpoznającego przedstawiono na rys. 4. Podstawowymi dziedzinami systemu są: scena, dziedzina scyfrowanych danych czujnikowych, dziedzina modelowania oraz pośredni symboliczny opis sceny (ang. *intermediate symbolic scene description*). Związki między tymi dziedzinami są następujące:
O – proces obrazowania (formowania obrazu) tworzy intensywnościowe lub zasięgowe dane wejściowe (oparte wyłącznie na fizyce wyczuwania);
C – proces cechowania danych wejściowych wydziela z tych danych odpowiednie, użytkowo niezależne cechy;
M – proces modelowania wytwarza wzorce dla obiektów sceny (automatycznie), w wyniku rekonstrukcji obiektów danych wejściowych bądź ręcznie – z wykorzystaniem np. programów geometrycznego modelowania brył);
R – proces rozpoznawania wymaga algorytmu pasowania między wzorcami i danymi wejściowymi;
U – proces ujmowania (przeglądania) sceny generuje syntetyczne dane wejściowe z wzorców obiektu.

System wizyjny powinien być zdolny do sygnalizowania osobom nadzorującym system zrozumienia sceny (słownie i wizyjnie).

Formalizacja problemu relacyjnego pasowania

Istnieją różne techniki rozpoznawania obiektów: rozpoznawanie statystyczne, strukturalne, syntaktyczne i inne. W 1-dowolno-widokowych systemach trójwymiarowych MP stosuje się przede wszystkim rozpoznawanie strukturalne.

Aby strukturalnie rozpoznać trójwymiarowy obiekt z dwuwymiarowej projekcji, należy relacyjnie dopasować projekcję obiektu do wzorca obiektu. Dopasowanie to musi dokonywać się z użyciem jakichkolwiek cech, które mogą pojawić się na ekranie i musi spełniać równanie geometrii rzutowej czujnika obrazowego.

Niech równania geometrii rzutowej czujnika, opisującej przekształcenie punktu (x, y, z) obiektu w punkt $\langle x', y' \rangle$ obrazu, mają postać:

$$x' = f(x, y, z, t, \theta)$$
$$y' = g(x, y, z, t, \theta)$$

gdzie: $t(t_x, t_y, t_z)$ – położenie czujnika, $\theta(\theta_x, \theta_y, \theta_z)$ – jego orientacja. Typ funkcji f i g jest taki, że jednemu punktowi na obrazie $\langle x', y' \rangle$ może odpowiadać wiele punktów na obiekcie $\langle x, y, z \rangle$. Jeżeli jednak, można znaleźć na obrazie N punkty $\langle x', y' \rangle$, dla których są znane położenia N punktów $\langle x, y, z \rangle$ na trójwymiarowym obiekcie (N dla większości czujników wynosi 4), to jest możliwe rozwiązanie powyższych równań dla położenia $t(t_x, t_y, t_z)$ i orientacji $\theta(\theta_x, \theta_y, \theta_z)$ czujnika.

Fakt ten pozwala na pasowanie relacyjne za pośrednictwem czujnikowej projekcji, które określa odpowiedniość między punktami obrazu i obiektu. Punkty na obrazie muszą oczywiście być łatwo rozróżnialne. Mogą to być: środek plamy, punkty końcowe odcinka, kąty między przecinającymi się liniami itp., wykrywalne na obrazie za pomocą ogólnie znanych operacji wydzielania cech obrazu.

Można przyjąć następującą strukturę na obrazie:
V' – zbiór punktów obrazu, V – zbiór punktów obiektu
(1') U' – zbiór łatwo wykrywalnych punktów na obrazie ($U' \subseteq V'$),
(2') L' – zbiór etykiet sprzęganych z tymi łatwo wykrywalnymi punktami obrazu ($L' = \{l'_i\}$), np. $L' = \{$*jasna plama, ciemna plama, kąt między liniami*$\}$,
(3') P' – funkcja etykietowania punktów obrazu U' etykietami ze zbioru $L'(P' : U' \to L')$.
Rozróżnialne punkty na obrazie mogą pozostawać w określonej relacji z innymi rozróżnialnymi punktami obrazu przez inne wykryte cechy: łuki, linie brzegowe lub pozostawanie w tym samym jednorodnym obszarze.
(4') A' – zbiór relacji, które mogą istnieć między punktami obrazu

$(A'_i = \{a_i\})$, np. $A' = \{$*prostoliniowy brzeg, krzywoliniowy brzeg, prosta, krzywa, pozostawanie w tym samym jednorodnym obszarze*$\}$.

(5') R' – zbiór par rozróżnialnych punktów z U' etykietowanych relacyjnymi etykietami z A' ($R' \subseteq U' \times U' \times A'$).

Przy tych oznaczeniach ogólna struktura relacyjna na obrazie jest dana za pomocą piątki (U', L', A', P', R').

Podobny układ zależności można wprowadzić dla zbioru V trójwymiarowych punktów obiektu. Niech będą dane oznaczenia:

U – zbiór trójwymiarowych punktów rzutowanego obiektu, których rzuty na obrazie mają charakterystyczne cechy ($U \subseteq V$),
L' – zbiór etykiet dla punktów obiektu (typów rozróżnialnych cech), taki jak dla punktów obrazu,
P – relacje sprzężenia punktów z U z etykietą, którą mają odpowiadające im punkty na obrazie ($P \subseteq U \times L'$) (relacja P odgrywa w relacyjnej strukturze obiektu rolę analogiczną, jak relacja P' w relacyjnej strukturze obrazu, tj. P: $U \to L'$),
A' – zbiór relacji między punktami obiektu, taki sam jak dla punktów obrazu.

Zależnie od kierunku oglądania, kształtu obiektu i sposobu wyczuwania, pary punktów z U (tj. na obiekcie) mogą mieć odpowiadające im pary z U' (tj. na obrazie), pozostające ze sobą w jednej z relacji z A'.
R – zbiór wszystkich trójek (u_1, a', u_2) takich, że dla pewnych kierunków oglądania charakterystycznych punktów obiektu u_1 i u_2 są źródłem punktów na obrazie, pozostających między sobą w relacji a'.

Taka relacyjna struktura obiektu jest opisywana przez piątkę (U, L', A', P, R).

Pasowanie obiektów do wzorców za pośrednictwem rzutowania czujnikowego może być wówczas zdefiniowane w pojęciach relacyjnej zgodności między relacyjną strukturą obrazu (U', L', A', P', R') a relacyjną strukturą wzorca (U, L', A', P, R), przy założeniu spełnienia równań geometrii rzutowej czujnika między punktami z U' (na obrazie) i dobranymi do nich punktami z U (na obiekcie). Procedura pasowania musi określać pewną, możliwie największą część struktury obrazu (U', L', A', P', R'), która pasuje do struktury obiektu (U, L', A', P, R).

Oczywiście, pasowanie jest związane z poszukiwaniem. Aby było wydajniejsze, dzieli się problem na dwie części:
- fazę poszukiwania kandydata do pasowania,
- fazę potwierdzenia dopasowania.

W pierwszej fazie szuka się po wszystkich podzbiorach U'_p punktów charakterystycznych na obrazie. Można tak ukierunkować poszukiwania, aby dotyczyły jedynie podzbioru rozmiaru N, gdzie N jest najmniejszą liczbą tych punktów obrazu, które – gdy pasują do punktów obiektu – mogą gwarantować rozwiązanie zadania dla nieznanego położenia i orientacji czujnika.

Formalnie relacyjnym pasowaniem nazywa się funkcję $h: U'_p \to U$, gdy:
1) odpowiadające sobie punkty obrazu i obiektu są tego samego typu,
2) pary odpowiadających sobie punktów na obrazie i obiekcie mają tę samą relacyjną etykietę a'.

Kandydat do relacyjnego pasowania, znaleziony w wyniku poszukiwania musi zostać sprawdzony. Sprawdza się, czy istnieje położenie t i orientacja θ czujnika, przy których każdy punkt z U, odpowiadający charakterystycznemu punktowi z U'_p, spełnia ze względu na t i θ równania geometrii rzutowej czujnika:

$$x'(u) = f(x(u), y(u), z(u); t, \theta)$$
$$y'(u) = g(x(u), y(u), z(u); t, \theta)$$
$$u = h(u')$$

Jeżeli takie t i θ są znalezione dla określonego pasowania h, to jest wykonywana druga faza procedury pasującej; jeżeli nie, to nie ma podstaw do uznania odpowiedniości między punktami obrazu i punktami obiektu.

W drugiej fazie procedury pasującej usiłuje się znaleźć informację potwierdzającą odpowiedniość punktów obrazu i obiektu. Potwierdzenie to wymaga znalezienia dodatkowych dwuwymiarowych punktów i relacji nie wykorzystywanych w początkowym pasowaniu kandydata. Faza potwierdzenia różni się od fazy szukania kandydata tym, że nie ma w niej potrzeby potwierdzenia dopasowania ponieważ położenie i orientacja czujnika są znane.

Na etapie potwierdzenia punkt u ma trójwymiarowe współrzędne $(x(u), y(u), z(u))$. Ponieważ położenie t i orientacja θ czujnika są już znane, więc uwzględniając równania geometrii projekcyjnej czujnika otrzymuje się położenie punktu obrazu (x'^*, y'^*):

$$x'^* = f(x(u), y(u), z(u); t, \theta)$$
$$y'^* = g(x(u), y(u), z(u); t'\theta)$$

Punkt $\langle x'^*, y'^* \rangle$, oczywiście, może być lub może nie być punktem z charakterystyczną cechą obrazu. Jeśli jest, to istnieje pasowanie h takie, że

$$\langle x'^*, y'^* \rangle = \langle x'(u'), y'(u') \rangle$$

Funkcja h^*: $U'^* \rightarrow U^*$ zdefiniowana przez:

$$h^*(u') = u, \text{ gdy} \left| \begin{array}{l} x'(u') = f(x(u), y(u), z(u); t, \theta) \\ y'(u') = g(x(u), y(u), z(u); t'\theta) \\ (u, P'(u')) \in P, \end{array} \right.$$

ustala wszystkie możliwe pasowania kandydata między punktami na obrazie a punktami na obiekcie, które są zgodne z równaniami geometrii rzutowej czujnika i mają te same dopuszczalne typy etykiet.

Wyliczanie h^* jest bezpośrednie, jeśli tylko t i θ są znane. Dla każdego trójwymiarowego punktu $u \in U'$ należy określić projekcję i doglądać, czy w obszarze projekcyjnym jest dwuwymiarowy punkt $u' \in U'$. Jeśli tak, to tak, to należy sprawdzić typy ich etykiet i jeśli obydwa te punkty mają ten sam typ etykiet – dodać pasowanie do h^*

Mając wyliczone h^* można skompletować proces potwierdzenia. Świadectwo potwierdzające może być dane w części przez różnicę wielkości h^* i h, a w części – również przez liczbę relacyjnych pasowań h^*, które poza tym mogą utrwalać pasowania ustalone przez h.

Jeżeli funkcja h^* rozszerza pasowanie h na wiele zależności, to potwierdzenie jest znaczące, a rozszerzone pasowanie kandydatów h^* staje się pasowaniem między obiektem a obrazem.

Trudność pasowania wynika przede wszystkim stąd, że niektóre zależności obowiązujące w trójwymiarowym obiekcie nie muszą obowiązywać na dwuwymiarowym obrazie i na odwrót, pewne zależności obowiązujące na obrazie nie muszą obowiązywać w trójwymiarowym obiekcie. A więc całkowite pasowanie obrazu do modelu obiektu nie jest rozwiązaniem modelu pasowania jako takiego.

Charakterystyka idealnego systemu rozpoznającego

Nie twierdzę, oczywiście, że jakiś istniejący system MP ma wszystkie te możliwości ani, że najpilniejszym zadaniem w systemach wizyjnych jest zbudowanie takiego właśnie systemu. Motywem moim jest opisanie działania pewnego systemu rozpoznającego ogólnego przeznaczenia, który by spełniał większość wymagań stawianych przez badaczy systemów MP. Idealny system MP powinien mieć następujące własności:

- musi wykorzystywać dane wejściowe z dowolnego punktu obserwacji, tj. bez wyróżniania określonego, np. poziomego kierunku,
- musi radzić sobie z dowolnie złożonymi obiektami sceny bez preferowania pewnych (np. płaskich) powierzchni,
- musi pokierować dowolną kombinacją dostatecznie dużej liczby obiektów w dowolnym położeniu i orientacji,
- musi radzić sobie z pewnym poziomem szumu w danych wejściowych, bez zmniejszenia swoich możliwości,
- powinien analizować scenę szybko i poprawnie,
- powinien móc zmodyfikować dane modelu sceny, by pokierować wykrywaniem nowych obiektów bądź sytuacji,
- jest pożądane, aby mógł wyrażać własną pewność w jego zrozumieniu danych wejściowych.

LITERATURA

[1] Besel P. J., Ramesh J. C.: Three-dimensional object recognition. Computing Survays, Vol. 17, No. 1, pp. 75–145, 1985
[2] Falk G.: Interpretation of imperfect line data as three-dimensional scene. Artificial Intelligence, Vol. 3, No. 2, pp. 101–144, 1972
[3] Haralick R. M.: Computer vision theory: the lack thereof. Computer Vision, Graphics, and Image Processing, No. 36, pp. 372–386, 1986
[4] Ikeuchi K., Horn B. K. P.: Numerical shape from shading and occluding boundaries. Artificial Intelligence, Vol. 17, No. 1–3, pp. 141–184, 1981
[5] Pozdniak G. E. (Ed.): Integralnye roboty. Sbornik statiej. Mir, Moskva 1975
[6] Roberts L. G.: Machine perception of three-dimensional solids. Optical and Electro-Optical Information Processing. Tippet J. T. et al. (Eds). MIT Press, Cambridge (MA), pp. 159–197, 1963
[7] Shirai Y. : Three-dimensional computer vision. Springer-Verlag, Berlin 1987.

abaQ

PRZEDSIĘBIORSTWO ZASTOSOWAŃ ELEKTRONIKI

ul. Obywatelska 128/152, 94–104 Łódź, tel. 86-49-30

oferuje:

- Płytki drukowane jedno- i dwustronne z metalizacją;
- Wykonywanie klisz i projektów płytek drukowanych;
- Programatory EPROM, PROM, 8748 do IBM PC/XT/AT;
- Emulatory EPROM;
- Asemblery i deasemblery skrośne procesorów 8-bitowych;
- Analizatory układów logicznych, 32-kanały, 20 i 40 MHz;
- Falowniki trójfazowe 2,2 kW, zasilanie 1- lub 3-fazowe, regulacja kierunku i prędkości do 200%, wejścia sterujące.

0/14/90

ZAKŁAD KOMPUTEROWYCH SYSTEMÓW POMIAROWYCH
ul.Zb. z Bogdańca 4
80-419 Gdańsk

digimer tel. 419519
teleks 512290

Polecamy usługi w zakresie:

- automatyzacji pomiarów, zbierania i przetwarzania danych pomiarowych w oparciu o komputery PC XT/AT
- konstrukcji testerów produkcyjnych dla urządzeń elektronicznych cyfrowych i analogowych,
- realizacji interfejsów pomiarowych z oprogramowaniem,
- dostaw sprzętu komputerowego.

Oferujemy wyposażenie PC XT/AT z oprogramowaniem:

☆ PROGRAMATORY PAMIĘCI (E)EPROM (16 kb – 2 Mb);
☆ PROGRAMATORY PAMIĘCI BPROM, PAL, PEEL, GAL, EPLD, FPL;
☆ PROGRAMATORY μP JEDNOUKŁADOWYCH;
☆ SYMULATORY PAMIĘCI EPROM (16 kb – 512 kb, 8 i 16 bit);
☆ TESTERY CYFROWYCH UKŁADÓW SCALONYCH – dla TTL, CMOS i RAM, – autoidentyfikacja, edytor i debuger wzorca;
☆ WIELOKANAŁOWE PRZETWORNIKI A/D 12 bit, od 40 μs,
– z separacją galwaniczną preferowane do zastosowań w przemyśle, medycynie i na stanowiskach badawczych;
☆ KARTY PROTOTYPOWE PC/XT (dł. 32,5 cm);
☆ ANALIZATORY STANÓW LOGICZNYCH do 100 MHz – nowość;
☆ KASOWNIKI PAMIĘCI EPROM (X10) – czas kasowania 15 s.

Posiadamy w ciągłej sprzedaży pamięci RAM i EPROM.

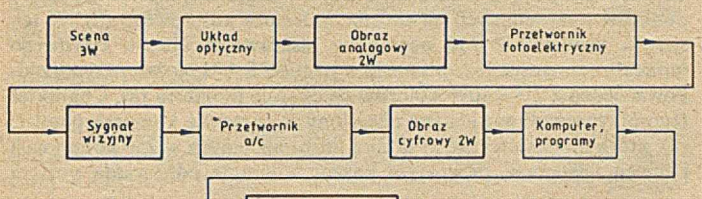
0/13/90

ANDRZEJ MATERKA
WŁODZIMIERZ GÓRNISIEWICZ
MARIUSZ KUKUŁA
Instytut Elektroniki
Politechnika Łódzka

Mikrokomputerowe systemy przetwarzania obrazów

Wiele badań wykonywanych w placówkach ochrony zdrowia, pracowniach naukowych oraz przemysłowych laboratoriach kontroli jakości obejmuje analizę informacji zawartej w obrazach, na przykład mikroskopowych lub rentgenowskich. Pierwsze próby wykorzystania komputerów jako narzędzi wspomagających te prace przeprowadzono na początku lat pięćdziesiątych. Poza zastosowaniami wojskowymi, dotyczyły one głównie analizy obrazów komórek i przekrojów tkanek w badaniach medycznych. Próby te wykazały możliwość obiektywizacji i skrócenia czasu badań w porównaniu z metodami tradycyjnymi. Dały też początek nowej, interesującej gałęzi wiedzy, przyciągającej specjalistów z różnych dziedzin nauki i techniki.

Typowy przebieg procesu przetwarzania informacji obrazowej przedstawiono na rys. 1. Rozkład jaskrawości (luminancji) badanej sceny trójwymiarowej (3W) jest rzutowany przez układ optyczny na płaszczyznę fotoczułą przetwornika fotoelektrycznego, np. w kamerze telewizyjnej. Powstaje w ten sposób analogowy obraz dwuwymiarowy (2W), będący sygnałem wejściowym dla przetwornika. Rozkład jaskrawości tego obrazu przetwarza się na sygnał wizyjny, który jest próbkowany, kwantowany i zapamiętywany jako dwuwymiarowy obraz cyfrowy. Obraz cyfrowy jest w istocie uporządkowanym zbiorem liczb, wykorzystywanym w obliczeniach mających na celu wyznaczenie cech informacyjnych sceny trójwymiarowej (np. kształtu obiektów sceny).



Rys. 1. Schemat przetwarzania informacji obrazowej

Cechą szczególną przetwarzania obrazów jako działu informatyki i elektroniki jest relatywnie duża liczba bitów pamięci niezbędna do cyfrowego zapisu jaskrawości obrazu. Sięga ona, na przykład, pół miliona bajtów dla pojedynczego, nieruchomego obrazu telewizji monochromatycznej. Cała pamięć RAM komputera IBM PC/AT w typowej konfiguracji wystarcza do zapamiętania zaledwie jednego takiego obrazu, bez możliwości zapisu skromnego choćby programu analizy zarejestrowanej informacji. Przetwarzanie danych tego rodzaju wymaga komputerów o dużej szybkości działania, z pamięcią o dużej pojemności i wieloma procesorami pracującymi jednocześnie w przypadku obliczeń w czasie rzeczywistym. W większości zastosowań źródłem danych jest przetwornik analogowo-cyfrowy, zastosowany do próbkowania i kwantowania sygnału z kamery telewizyjnej. Szerokość pasma sygnału wizyjnego wyznacza bardzo małe odstępy czasowe między kolejnymi próbkami – rzędu 80 ns. Określa to z kolei rodzaj wymaganego przetwornika a/c (przetwornik równoległy o odpowiednio krótkim czasie konwersji) i pamięci RAM obrazu (pamięć o krótkim czasie dostępu i dużej pojemności). Wszystkie te czynniki decydują o ogromnych kosztach systemów cyfrowego przetwarzania obrazów. Sprzęt taki był (i jest nadal) dostępny w nielicznych ośrodkach, co ogranicza możliwości powszechniejszego wykorzystania cyfrowych metod analizy obrazów w praktyce. Niedostateczne rozpowszechnienie środków sprzętowych ogranicza również rozwój metod przetwarzania obrazów. Wynika to między innymi z braku możliwości doświadczalnej weryfikacji algorytmów.

Nowe szanse rozwoju tej ważnej dziedziny stworzyła sytuacja z końca lat osiemdziesiątych, w wyniku której tak zwane komputery osobiste mają moce obliczeniowe porównywalne z dużymi komputerami z końca lat siedemdziesiątych. Jednocześnie, coraz powszechniejsze wykorzystanie komputerów typu IBM PC/XT, AT, 386 do prac profesjonalnych w różnych dziedzinach gospodarki uzasadnia celowość opracowywania środków sprzętowych i programowych umożliwiających zastosowanie tych urządzeń do przetwarzania obrazów. W praktyce okazuje się, że komputer klasy IBM PC/AT, wyposażony w odpowiednie układy rozszerzające i programy, może być narzędziem wspomagającym analizę obrazów o niezbędnej dla większości profesjonalnych zastosowań rozdzielczości 512×512 punktów obrazu cyfrowego (piksli). Czas wykonywania typowych zadań przetwarzania obrazów (np. filtracji w celu wydzielenia krawędzi obiektów w obrazie) w takim systemie jest rzędu kilkunastu sekund, co całkowicie wystarcza w badaniach laboratoryjnych i pracach naukowych.

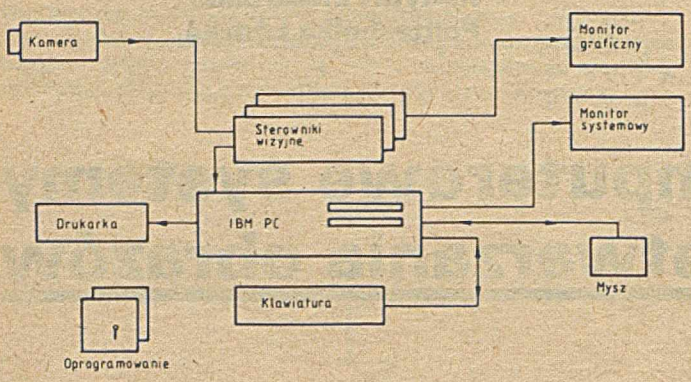
Celem tego artykułu jest przedstawienie zasadniczych wyników prac z dziedziny cyfrowych metod przetwarzania obrazów, prowadzonych od 1985 r. w Instytucie Elektroniki Politechniki Łódzkiej, na mikrokomputerach IBM PC. Scharakteryzowano również tendencje rozwoju mikrokomputerowych systemów przetwarzania obrazów (MSPO). Opisane prace zostały wykonane w dziesięcioosobowym Zespole Elektronicznej Aparatury Medycznej (EAMD); zapoczątkowało je opracowanie analizatora obrazów o rozdzielczości 128×128 piksli, z wykorzystaniem mikroprocesora Motorola 6800, a następnie zaowocowały one oryginalnymi projektami sterowników w standardzie IBM, umożliwiającymi zapamiętywanie i przetwarzanie obrazów o rozdzielczości kolejno 256×256 piksli (1987 r.) i 512×512 piksli (1989 r.). Specjalność dydaktyczna Zespołu miała wpływ na wybór przykładów zastosowań – z przewagą medycznych nad przemysłowymi.

SPRZĘT

Schemat funkcjonalny systemu o rozdzielczości 512×512 piksli, noszącego handlową nazwę MIPS-512, przedstawiono na rys. 2. Standardowy zestaw urządzeń IBM PC uzupełniono opracowanymi w Zespole specjalizowanym sterownikami, zainstalowanymi w gniazdach szczelinowych płyty głównej, wewnątrz obudowy komputera. Zainstalowanie sterowników nie ogranicza zakresu funkcji użytkowych, typowych dla komputerów IBM PC. Sterowniki umożliwiają cyfrową rejestrację obrazów oraz wyświetlania obrazów cyfrowych przetworzonych za pomocą komputera na ekranie monitora monochromatycznego i/lub kolorowego. Dodatkowo, układy elektroniczne sterowników umożliwiają modyfikowanie obrazów, przy użyciu znacznika sterowanego myszą.

Źródłem sygnału wizyjnego jest tu kamera telewizji monochromatycznej, a ogólniej – dowolne źródło całkowitego sygnału wizyjnego, według norm CCIR/OIRT. Kamera może być wyposażona w obiektyw

standardowy, może też być połączona z mikroskopem optycznym. Jest możliwe przystosowanie sterowników do współpracy z innym źródłem sygnału obrazu, np. z mikroskopem skaningowym lub kamerą termowizyjną.

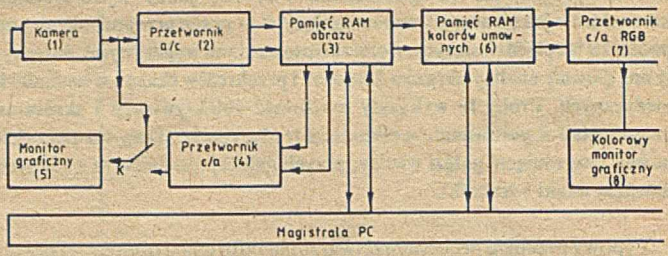


Rys. 2. Schemat funkcjonalny Mikrokomputerowego Systemu Przetwarzania obrazów MIPS – 512

Obraz cyfrowy ma postać kwadratowej tablicy o wymiarach 512 × 512 piksli, utworzonej przez próbkowanie i kwantowanie sygnału wizyjnego w chwilach odpowiadających środkowemu położeniu tablicy na tle analogowego obrazu telewizyjnego (rys. 3a). Jaskrawość punktów obrazu, wyznaczona wartościami próbek sygnału wizyjnego, jest kodowana z użyciem słowa 7-bitowego, przy czym czas rejestracji całej tablicy (aktywizacji obrazu) jest równy czasowi wyświetlania jednego obrazu TV (40 ms). Obraz cyfrowy jest zapisywany w pamięci RAM o pojemności 256 KB, umieszczonej na jednym ze sterowników rozszerzających. Pamięć ta zawiera 512 × 512 słów 8-bitowych. Siedem mniej znaczących bitów każdego słowa służy do zapisu skwantowanych wartości próbek sygnału wizyjnego. Ósmy bit jest wykorzystywany do przechowywania dodatkowych informacji, niezależnie od informacji zakodowanej w pozostałych bitach.

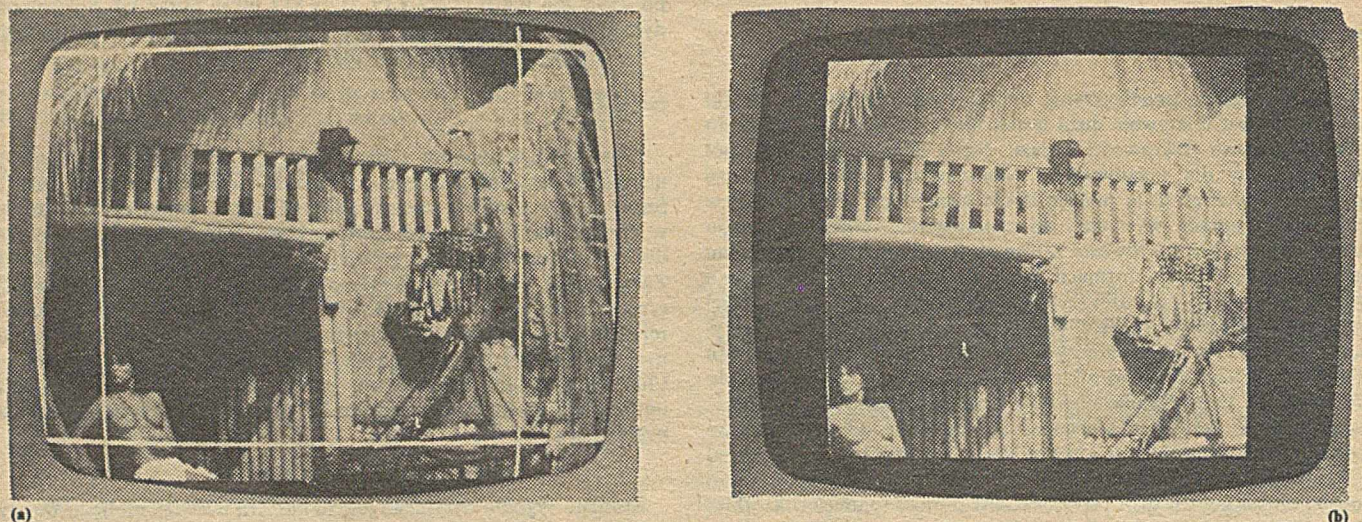
Punktom obrazu przyporządkowano układ współrzędnych Oxy z początkiem w lewym górnym rogu obrazu, przy czym piksel jest scharakteryzowany wartościami współrzędnych i jaskrawością. Każda ze współrzędnych przyjmuje wartości całkowite z przedziału $\langle 0, 511 \rangle$, jaskrawość punktu jest natomiast określona zawartością odpowiadającego mu słowa dwójkowego $b_0 b_i \ldots b_7$ pamięci obrazu. Dla $b_7 = 0$ jaskrawość przyjmuje jeden ze 128 poziomów danych słowem $b_0 b_1 \ldots b_6$. Dla $b_7 = 1$ jaskrawość punktu jest dodatkowo zwiększana o pewną wartość. Przyjętą organizację pamięci RAM obrazu można interpretować tak, że sterowniki wizyjne zawierają pamięć dwóch obrazów: obrazu z gradacją jaskrawości (GRAY) zapisanego w bitach $b_0 b_1 \ldots b_6$ i obrazu dwupoziomowego (OVER) zapisanego w bitach b_7 poszczególnych słów pamięci. Obraz OVER może być wyświetlany na tle obrazu GRAY, bez zmiany treści tego drugiego – niejako w drugiej płaszczyźnie ekranu monitora, zwanej płaszczyzną napisów lub nakładką. Daje to możliwość umieszczania napisów na ekranie, zaznaczania konturu obiektów i modyfikowania obrazów binarnych, np. za pomocą myszy i zestawu procedur grafiki komputerowej.

Schemat blokowy, ilustrujący współpracę kamery i monitorów z komputerem za pośrednictwem sterowników wizyjnych, przedstawiono na rys. 4. Sygnał wizyjny z kamery TV (1) pobudza wejście przetwornika a/c (2). W górnym położeniu klucza K wyjście kamery jest połączone także z wejściem monitora graficznego (5), co umożliwia obserwację obrazu „widzianego" przez kamerę. Jednocześnie, odpowiednie układy elektroniczne, nie pokazane na rysunku, wytwarzają sygnały generujące obraz kodu, wyznaczającego położenie obrazu cyfrowego na tle obrazu z kamery (por. rys. 3a). Wydanie odpowiedniego polecenia klawiatury IBM inicjuje proces próbkowania i kwantowania sygnału wizyjnego za pomocą przetwornika (2), z zapisem każdej skwantowanej próbki w bitach $b_0 b_1 \ldots b_6$ poszczególnych słów pamięci RAM (3), przy podstawieniu $b_7 = 0$. Po zakończeniu zapisu, klucz K jest przełączany w dolne położenie i na ekranie monitora pojawia się obraz cyfrowy, odpowiadający zawartości pamięci RAM (por. rys. 3b). Sygnał wizyjny, pobudzający wejście monitora (5) jest wytwarzany za pomocą przetwornika c/a (4) i odpowiedniego układu sterującego, nie uwidocznionego na rysunku.



Rys. 4. Uproszczony schemat blokowy sterowników wizyjnych

Sterowniki wizyjne zawierają także zestaw trzech 4-bitowych przetworników c/a RGB (7) i pamięć kolorów umownych (6). Wyjścia R, G, B przetworników (7) połączono z odpowiednimi wejściami kolorowego monitora graficznego (8). Kolejno odczytywane z pamięci RAM słowa $b_0 b_1 \ldots b_7$ określające jaskrawość poszczególnych punktów obrazu cyfrowego pobudzają wejścia adresowe pamięci kolorów umownych (6), przy czym każdemu adresowi odpowiada słowo 12-bitowe $r_0 r_1 r_2 r_3 g_0 g_1 g_2 g_3 b_0 b_1 b_2 b_3$. Bity r służą do kodowania jaskrawości składowej czerwonej, bity g – zielonej, a bity b – składowej niebieskiej barwy punktów wyświetlanych na ekranie monitora kolorowego (8). W ten sposób każdemu z 256 poziomów jaskrawości obrazu cyfrowego monochromatycznego zapisanego w pamięci RAM można przyporządkować jeden z $2^{12} = 4096$ kolorów na ekranie monitora (8). Charakter tego przyporządkowania jest określony zawartością komórek pamięci (6), którą można uważać za tablicę liczb zapisanych w 256 kolumnach i trzech wierszach, przy czym każdej kolumnie odpowiada poziom



(a) (b)

Rys. 3. Obraz z kamery telewizyjnej z zaznaczonym obszarem próbkowania (a); zarejestrowany obraz cyfrowy (b)

jaskrawości obrazu monochromatycznego, a kolejnym wierszom – jaskrawości składowych *R, G, B*, obrazu kolorowego. Podczas wyświetlania kolejnych punktów obrazu tablica ta służy do wyznaczania składowych barwy danego punktu, przypisanej jego jaskrawości (ang. *look-up table*). Sterowniki systemu MIPS-512 zawierają 8 takich tablic (palet kolorów umownych), które mogą być modyfikowane przez użytkownika za pomocą odpowiedniego programu komputerowego. Można dynamicznie zmieniać palety w czasie rzeczywistym – w celu uzyskania zamierzonych efektów barwnych.

Oprogramowanie

Równolegle z pracami projektowymi obejmującymi opracowanie modeli i prototypów sterowników przetwarzania obrazów, prowadzono w zespole EAM prace nad rozwojem metod teoretycznych przetwarzania obrazów i rozwojem oprogramowania systemu. W zgodzie z określonymi zadaniami dydaktycznymi Zespołu, wynikającymi z profilu kształcenia studentów w PŁ, podjęto wiele pomyślnych prób specjalizacji oprogramowania w trzech głównych dziedzinach zastosowań:
- wspomaganie diagnostyki medycznej.
- kształcenie studentów kierunku Elektroniczna Aparatura Medyczna,
- wybrane zastosowania przemysłowe.

Przy każdej adaptacji urządzenia do potrzeb konkretnego zastosowania nawiązywano bardzo ścisłą współpracę merytoryczną z odbiorcą pracy, w praktyce rozszerzając zespół badawczy o osobę przyszłego użytkownika systemu. W takim interdyscyplinarnym zespole formułowano funkcję, jaką system przetwarzania obrazów ma spełniać, ustalono warunki rejestracji obrazów, definiowano metody przetwarzania, określano formę wyprowadzenia wyników oraz sposób obsługi systemu. Po opracowaniu i ewentualnej weryfikacji doświadczalnej stosowanej metody analizy obrazu przygotowano program komputerowy umożliwiający praktyczną realizację postulowanej funkcji systemu. Opracowany program podlegał następnie wszechstronnym testom i modyfikacjom, z udziałem użytkownika. Przetestowane programy dołączano następnie do tworzonej w ten sposób biblioteki procedur przetwarzania obarzów.

Przez ponad pięć lat pracy nad rozwojem oprogramowania badano walory użytkowe dwóch różnych sposobów komunikacji użytkownika z komputerem. W pierwszym systematycznym podejściu, dotyczącym systemu o rozdzielczości 256 × 256 piksli, opracowano język przetwarzania obrazów CLIP (ang. *Computer Language for Image Processing*) [4]. Język ten, o cechach interpretatora, umożliwia wykonywanie operacji na obrazach traktowanych jako zmienne. Przykładowo, ciąg symboli języka CLIP

SHARPEN [KNIFE.1] [RAZOR.1]

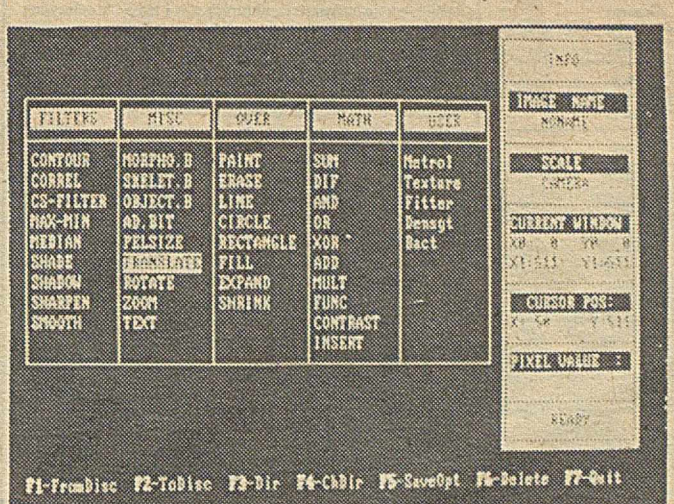
oznacza polecenie wykonania filtracji górnoprzepustowej obrazu o nazwie KNIFE.1 (uprzednio zdefiniowanego), utworzenie zmiennej obrazowej RAZOR.1 i przesłanie wyników filtracji do obszaru pamięci przyporządkowanego zmiennej RAZOR.1. Język CLIP umożliwia zapisanie ciągu operacji na obrazach, a więc tworzenie specjalizowanych programów przetwarzania obrazów. Z drugiej strony, opracowanie takich programów wymaga od użytkownika pewnej wiedzy ogólnej z programowania, co nie zawsze jest spełnione. Ponadto, formalizm języka CLIP, wymuszający konieczność wprowadzania z klawiatury nazwy polecenia oraz jego argumentów, wydłuża czas oczekiwania na wynik zaplanowanej operacji i, ogólnie, utrudnia komunikację użytkownika z systemem w trybie interakcyjnym. Cechy te należy zaliczyć do wad interpretatora. Drugie podejście, również przetestowane w systemie 256 piksli, polega na wykorzystaniu systemu rozwijanych okien informacyjnych, z opcjami odpowiadającymi poszczególnym operacjom na obrazach (ang. *pull-down menu*). Wydanie polecenia wykonania wybranej operacji sprowadza się do naciśnięcia jednego klawisza (i ewentualnie wcześniejszego naciśnięcia dwóch lub trzech innych – w celu wybrania odpowiedniego okna). Proces przygotowania systemu do wykonania polecenia ulega tu znacznemu skróceniu, co wielokrotnie sprawdzono korzystając z programu przetwarzania obrazu SHELL 256 [17], zaprojektowanego zgodnie z koncepcją rozwijanych okien. Wadą takiego podejścia, obserwowaną również w języku CLIP, jest brak informacji o pełnych możliwościach systemu przetwarzania obrazów. Informacja ta jest „ukryta" w oknach nie wyświetlonych albo znajduje się w podręczniku użytkownika i jej uzyskanie wymaga podjęcia specjalnych działań. Wydłuża to proces przygotowania systemu do pracy, zwłaszcza w przypadku procedur rzadko używanych.

Obecnie opracowywany program o nazwie SHELL 512, uruchomiony w MSPO 512 × 512 piksli, jest pozbawiony zasadniczych wad programów poprzednich. Założono, że ma on zapewnić prostotę obsługi systemu, przy czym ekran monitora systemowego IBM ma zawierać w czasie pracy systemu praktycznie pełną informację o dostępnych możliwościach przetwarzania obrazów. Postulowano ograniczenie liczby alternatywnie wyświetlonych okien informacyjnych do minimum, grupując odpowiednio polecenia programu. Oczywiście nie istnieje ogólne rozwiązanie tak postawionego zadania optymalizacji programu. Potrzebne są pewne dodatkowe założenia, związane z częstością wywoływania poszczególnych poleceń w przewidywanych zastosowaniach systemu i przewidywaną wiedzą użytkownika w dziedzinie metod przetwarzania obrazów oraz podstaw informatyki. Założono mianowicie, że głównymi użytkownikami systemu są:
- pracownicy laboratoriów medycznych i biologicznych,
- studenci studiów technicznych,
- projektanci systemów przetwarzania obrazów.

W pierwszym przypadku system ma stanowić narzędzie ułatwiające badania biomedyczne osobie, która nie ma czasu na wprowadzanie skomplikowanych poleceń z klawiatury. Na przykład, do wykonania ciągu operacji mających na celu zliczenie kolonii bakteryjnych wyrosłych na płytce Petriego powinno wystarczyć naciśnięcie jednego klawisza, po uprzednim ustawieniu płytki w polu widzenia kamery. W przypadku drugim założono, że student w laboratorium przetwarzania obrazów powinien mieć możliwość samodzielnego składania ciągu elementarnych operacji w rodzaju: rejestracja obrazu, wyrównywanie tła, segmentacja przez progowanie i zliczanie obiektów w obrazie binarnym po segmentacji. Ponadto, dyplomanci specjalizujący się w dziedzinie metod przetwarzania obrazów powinni mieć możliwość dołączania własnych modułów programowych do udostępnionej im biblioteki skomplikowanych procedur, z możliwością wywołania tak opracowanych procedur z głównego menu programu. Podobne możliwości powinny być udostępnione projektantom systemów przetwarzania obrazów.

Program SHELL 512, spełniający powyższe postulaty, obejmuje następujące grupy poleceń przetwarzania i przesyłania informacji obrazowej:
- polecenia systemowe (komunikacja z MS-DOS, komputerem i sterownikami),
- biblioteka standardowych poleceń przetwarzania obrazów,
- polecenia zdefiniowane potrzebami użytkownika.

Nazwy większości często używanych procedur są wyświetlone na ekranie monitora przez cały czas pracy systemu. Ponadto, w oknie informacyjnym są wyświetlone komunikaty o stanie systemu (współrzędne aktywnego okna obrazowego, nazwa ostatnio załadowanego pliku itp.). Wykonanie procedury musi być poprzedzone wyświetleniem jej nazwy i naciśnięciem klawisza *enter* lub tylko naciśnięciem wyróżnionego klawisza. Polecenia systemowe, standardowe polecenia przetwarzania obrazów i fotografię menu głównego przedstawiono w tabelach 1 i 2 oraz na rys. 5.



Rys. 5. Menu główne programu SHELL 512

Przykłady zastosowań

Omówiony MSPO w postaci handlowej, obejmującej sprzęt i oprogramowanie, opracowano w ramach programu centralnego finansowania prac badawczych wdrożeniowych CPBR 8.6. Metody przetwarzania obrazów będące podstawą do projektowania programów, obejmujące modelowanie elementów i układów elektronicznych systemu, identyfikację głównych źródeł zniekształceń informacji obrazowej i opracowanie metod redukcji zniekształceń rozwijano w ramach programu badań podstawowych CPBP 02.14. Opracowane systemy wdrożono do produkcji w SZTW „Omega" w Łodzi (system 256 × 256 piksli) oraz w PZ RODAN w Łodzi (system 512 × 512 piksli). Ogółem do końca I połowy 1990 r. skonstruowano i zainstalowano u użytkowników w Polsce i ZSRR 15 mikrokomputerowych systemów przetwarzania obrazów, według projektów Zespołu Elektronicznej Aparatury Medycznej.

Program METROL

W wielu zastosowaniach MSPO pełni funkcję narzędzia do pomiaru wielkości geometrycznych, np. odległości między wybranymi punktami obrazu. Dokładność pomiaru w każdym przypadku zależy od dokładności skalowania systemu, to jest przeliczenia długości wzorca, wyrażonej przykładowo w milimetrach, na liczbę piksli obrazu cyfrowego. Tradycyjnie MSPO są skalowane przy użyciu myszy lub dyskretyzatora, z dokładnością wyznaczoną liczbą punktów obrazu. Jeśli, na przykład, obraz w postaci tablicy 512 × 512 piksli nie zapewnia dostatecznej dokładności, to rozważa się zwykle użycie systemu o większej liczbie piksli, powiedzmy 1024 × 1024, z kamerą o stosownie większej rozdzielczości. W efekcie zwiększenie dokładności skalowania i pomiarów powoduje zwiększenie kosztu systemu i wydłuża czas przetwarzania obrazów.

Odmienne, oryginalne podejście zastosowano w programie METROL. Dobierając odpowiednio funkcję matematyczą opisującą „rozmycie" brzegów wzorca długości w obrazie z typowej kamery, uzyskano zmniejszenie średniokwadratowego błędu skalowania o rząd wielkości, w porównaniu z metodą tradycyjną [12]. System o rozdzielczości 512 × 512 piksli, z kamerą konwencjonalną, daje przy użyciu programu METROL możliwość skalowania i pomiaru z dokładnością odpowiadającą systemowi o rozdzielczości ok. 5000 × 5000 piksli.

Program VEIN

Program ten umożliwia pomiar parametrów geometrycznych naczyń krwionośnych (długość, średnica, pojemność, pole powierzchni tkanek objętych działaniem wybranego układu naczyń) [3, 5, 9, 10, 11]. Zastosował go dr med. Jerzy Gielecki, z Akademii Medycznej w Bydgoszczy, do badań rozwoju koła tętniczego mózgu, w ramach dysertacji doktorskiej.

Program BACT

Program BACT służy do zliczania kolonii bakteryjnych, np. hodowanych na płytkach Petriego. Automatyczna analiza kształtu poszczególnych obiektów obrazu umożliwia rozdzielenie tych kolonii, które połączyły się podczas wzrostu (rys. 11). Rozróżnienie kształtów daje też możliwość oddzielnego badania różnych szczepów bakteryjnych, hodowanych na wspólnym podłożu. Program BACT został opracowany

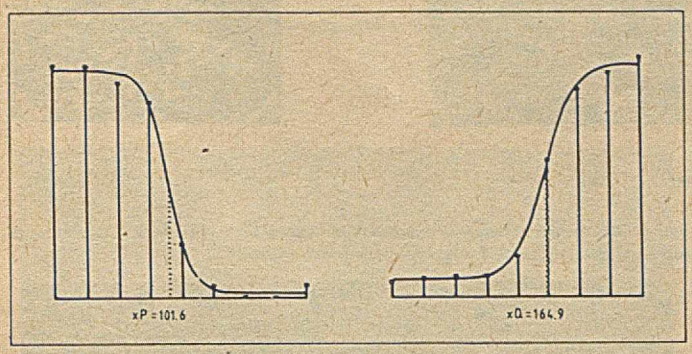
Tabela 1. Polecenia systemowe

Polecenie	Znaczenie
FromDisc	Przeniesienie obrazu cyfrowego z pamięci masowej do pamięci operacyjnej
ToDisc	Zapisanie obrazu cyfrowego wyświetlonego na monitorze graficznym w pamięci masowej
Dir	Wyświetlenie katalogu systemu MS-DOS
ChDir	Zmiana katalogu
Delete	Skasowanie pliku z katalogu w pamięci masowej
Quit	Zakończenie pracy programu SHELL 512
Scan	Cyfrowa rejestracja obrazu
Invert	Inwersja poziomów jasności obrazu
Histo	Wykreślenie histogramu obrazu na monitorze systemowym
Plot	Wykreślenie poziomów jaskrawości obrazu w rzucie aksonometrycznym
Level	Wyświetlenie liczby określającej poziom jaskrawości piksla wskazanego za pomocą myszy
Window	Definiowanie fragmentu obrazu do dalszego przetwarzania
Scale	Skalowanie obrazu
Element	Definiowanie elementu strukturującego i maski korelacyjnej
Palette	Definiowanie palety pseudokoloru
Change	Zmiana parametrów (opcji) programu
SaveOpt	Zapisanie opcji programu w pamięci masowej
Restore	Przywrócenie początkowych wartości opcji programu
Clr.IMA Clr.IMB Clr.OVR Clr.IMG	Wpisanie do obrazu określonych poziomów jaskrawości („czyszczenie" obrazu)
Frame-B	Skopiowanie obrazu cyfrowego wyświetlonego przez monitor graficzny do pamięci pomocniczej
B-Frame	Skopiowanie obrazu cyfrowego z pamięci pomocniczej do pamięci RAM obrazu
IMB-OVR	Umieszczenie obrazu binarnego w płaszczyźnie napisów
OVR-IMB	Przeniesienie obrazu binarnego z płaszczyzny napisów do płaszczyzny obrazu
XCHG.IMA	Wymiana zawartości pamięci RAM obrazu i pamięci pomocniczej
XCHG.OVR	Wymiana płaszczyzny napisów między pamięcią RAM obrazu a pamięcią pomocniczą

Tabela 2. Standardowe polecenia przetwarzania obrazów

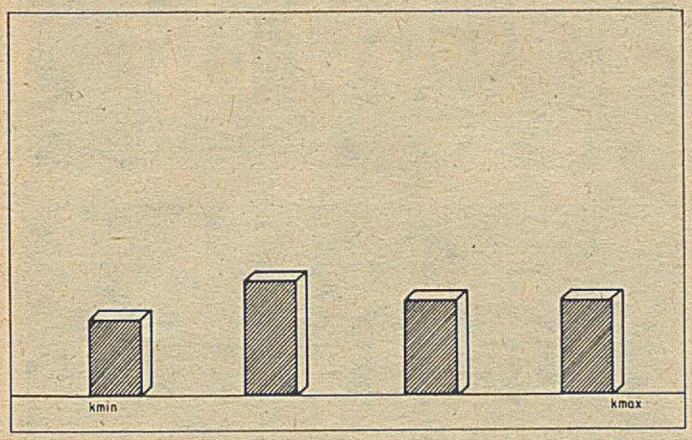
Polecenie	Znaczenie
CONTOUR	Procedury wydzielania konturu obiektów
CORREL	Korelacja obrazu z zadaną maską
CS-FILTER	Adaptacyjny filtr zakresowy
MAX-MIN	Filtr morfologiczny dla obrazów typu GRAY
MEDIAN	Filtry środkowe (medianowe)
SHADE	Wyrównywanie tła obrazu
SHADOW	Niesymetryczny filtr górnoprzepustowy
SHARPEN	Filtr górnoprzepustowy
SMOOTH	Filtry dolnoprzepustowe
MORPHO.B	Procedury morfologiczne dla obrazów binarnych
SKELET.B	Wyznaczanie linii szkieletowych
OBJECT.B	Rozpoznawanie obiektów obrazu
PELSIZE	Symulacja zmniejszonej rozdzielczości systemu
TRANSLATE	Przesunięcie fragmentu obrazu
ROTATE	Obrót fragmentu obrazu
ZOOM	Powiększenie lub zmniejszenie fragmentu obrazu
TEXT	Redagowanie tekstu w płaszczyźnie napisów
PAINT ERASE LINE CIRCLE RECTANGLE FILL EXPAND SHRINK	Modyfikowanie obrazów binarnych w płaszczyźnie napisów
SUM DIF AND OR XOR ADD MULT FUNC	Punktowe operacje matematyczne i logiczne na obrazach
CONTRAST	Procedury poprawiania kontrastu
INSERT	Warunkowe łączenie dwóch obrazów

z udziałem dra med. Eugeniusza Małafieja, z Zakładu Mikrobiologii WAM w Łodzi, w ramach wspólnych prac zmierzających do automatyzacji pracochłonnych i żmudnych badań mikrobiologicznych [7].

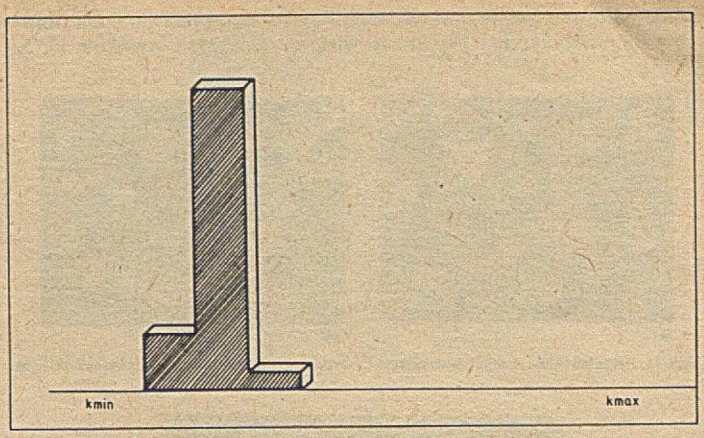


Rys. 6. Określenie położenia brzegów wzorca (linie przerywane) za pomocą funkcji opisującej rozmycie obrazu wynikłe ze skończonej rozdzielczości kamery. Wzorcem jest ciemny prostokąt na jasnym tle, o znanej długości boku równoległego do osi poziomej *Ox*. Rysunek przedstawia ciągi próbek obrazu wzdłuż linii poziomej, odpowiednio w pobliżu lewej i prawej krawędzi wzorca. Obliczona długość (164,9 — 101,6 = 63,3) może przyjmować wartości ułamkowe, co nie jest możliwe przy skalowaniu tradycyjnym, z użyciem arytmetyki całkowitoliczbowej

Rys. 7. Histogram pomiaru współczynnika skali *k* systemu z kamerą widikonową typu TPK-162, dla wzorca o długości 10 cm



a) skalowanie tradycyjne, δ = 0,027 cm/piksel,



b) skalowanie za pomocą programu METROL, δ = 0,003 cm/piksel
δ – błąd średniokwadratowy pomiaru współczynnika skali

Program OBJECT

Program OBJECT umożliwia obliczanie parametrów geometrycznych obiektów (np. obwód, pole powierzchni, średnice Fereta, kąt nachylenia osi głównej itd.), do późniejszej analizy statystycznej. Przykładem zastosowań takiej analizy są badania wpływu substancji toksycznych na zmiany morfologiczne w żywych tkankach, prowadzone w Instytucie Medycyny Pracy w Łodzi [4].

Program SHADE

W wielu zastosowaniach (biologia, medycyna, metalurgia, robotyka) obrazy cechuje nierównomierność oświetlenia, utrudniająca analizę informacji. Automatyczne wyrównywanie oświetlenia umożliwia program SHADE. W programie wykorzystano między innymi metodę opisaną w [14].

Program DENSIT

Program DENSIT służy do ilościowej analizy preparatów uzyskiwanych w procesie elektroforezy. System MIPS-512 wyposażony w program DENSIT spełnia funkcję łatwego w obsłudze i dokładnego densytometru.

Program FILTER

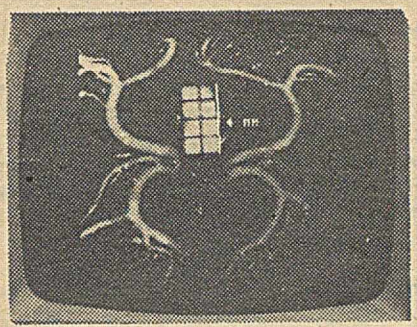
Zestaw procedur filtracji liniowej i nieliniowej umożliwia wstępne przetwarzanie obrazów w celu wydobycia i podkreślenia żądanych szczegółów obrazu, zawierających poszukiwane informacje.

Program WARP

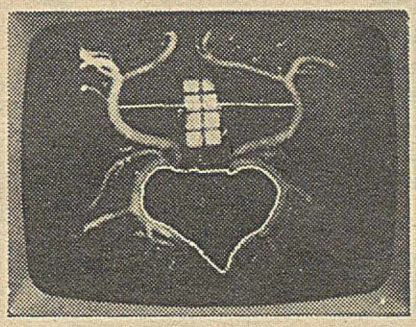
W wielu zastosowaniach obrazy ulegają zniekształceniom geometrycznym, których przyczynami są właściwości kamer (np. widikonowych) albo niedokładności układu optycznego (np. soczewek mikroskopu). W efekcie istnienia zniekształceń, wyniki pomiarów geometrycznych zależą od położenia obiektów w obrazie, co prowadzi do błędów. Automatyczną korekcję zniekształceń umożliwia program WARP, z użyciem metody opisanej w [12].

Perspektywy rozwoju

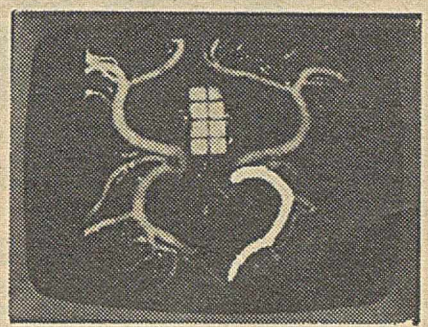
Zasadnicze wyniki przedstawionych prac w dziedzinie przetwarzania obrazów można uporządkować w kilku różnych aspektach. Przede wszystkim prace konstrukcyjne dały solidną bazę sprzętową do badań podstawowych i wdrożeniowych. Prace podstawowe były ukierunko-



Rys. 8. Obraz naczyń koła tęczowego mózgu (KTM) z naniesioną podziałką milimetrową do skalowania systemu

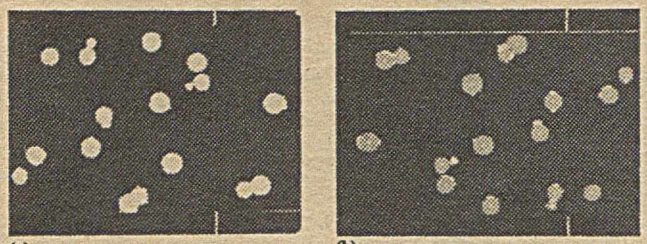


Rys. 9. Linie pomiaru średnicy części przedniej i pola powierzchni części tylnej KTM



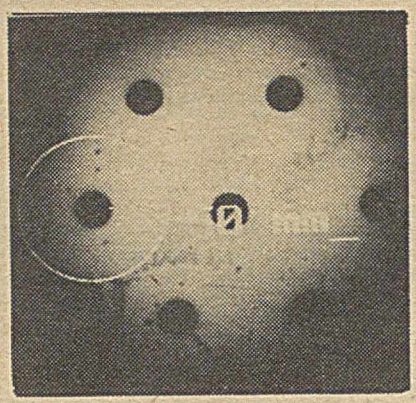
Rys. 10. Obraz KTM z zaznaczonym naczyniem, wybranym przez anatoma w celu przeprowadzenia pomiaru objętości (pojemności) naczynia

wane na opracowanie nowych metod cyfrowego przetwarzania obrazów biomedycznych i przyniosły wiele oryginalnych wyników [3, 5, 7–16].

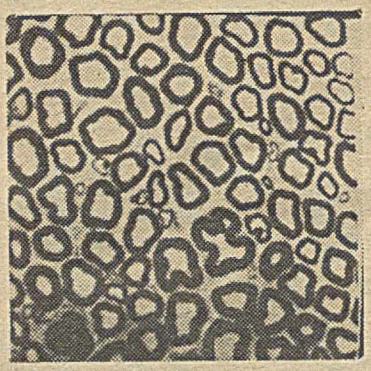


(a) (b)

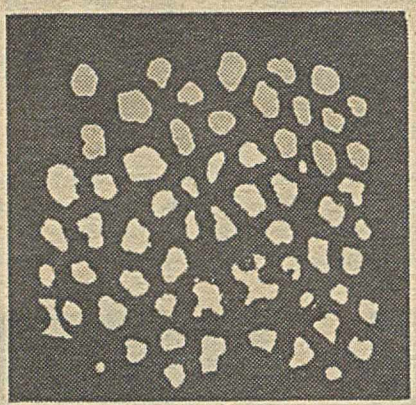
Rys. 11. Przykład obrazu kolonii bakteryjnych (a) przetworzonego za pomocą programu BACT (b)



Rys. 12. Pomiar średnicy stref zahamowania przy badaniu antybiotykoodporności

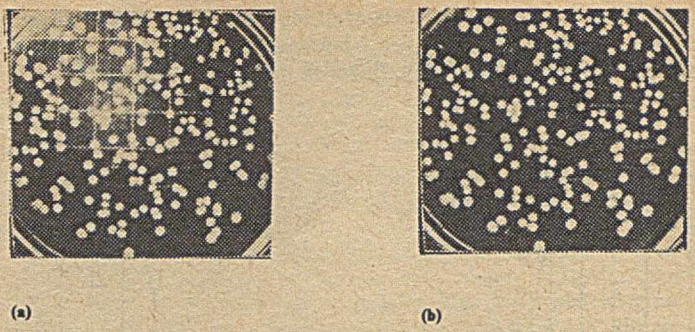


Rys. 13. Przekrój preparatu tkanki nerwowej



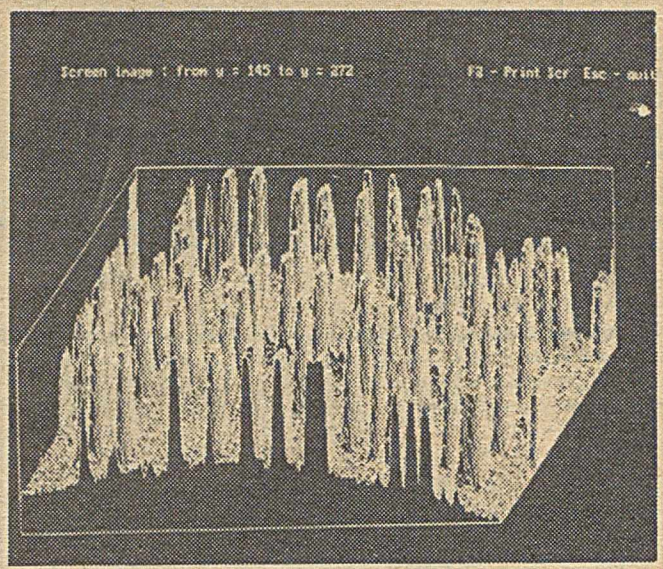
Rys. 14. Obraz z rys. 13 przetworzony za pomocą programu OBJECT. W zarejestrowanej na fotografii fazie przetwarzania zostały obliczone parametry geometryczne 16 obiektów w górnej części obrazu. Każdy z obiektów reprezentuje przekrój poprzeczny wnętrza odpowiednich włókien nerwowych

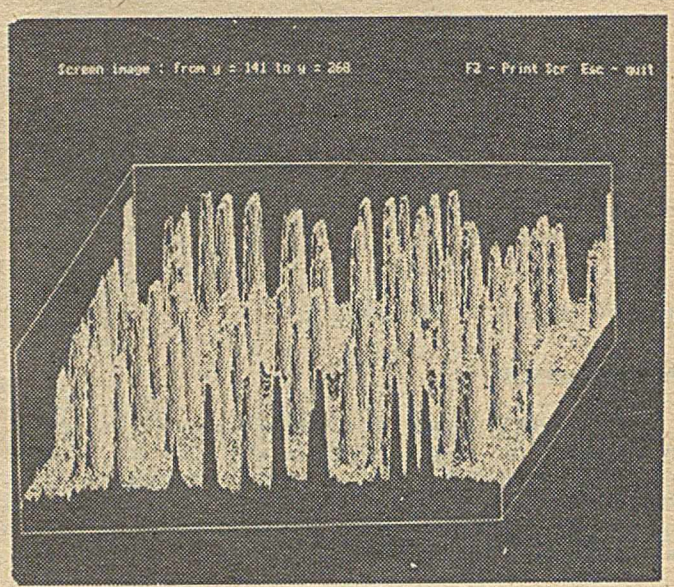
Nowy, ostatnio zapoczątkowany nurt tych badań dotyczy identyfikacji czynników ograniczających dokładność pomiaru cech obrazu w procesie przetwarzania i automatycznego rozpoznawania. Tworzy się modele przetworników fotoelektrycznych i układów wstępnego przetwarzania obrazu, uwzględniające wpływ takich czynników, jak skończona rozdzielczość kamery, szumy, zjawisko olśnienia, zniekształcenia



(a) (b)

Rys. 15. Obraz kolonii bakteryjnych wyhodowanych na płytce Petriego (a); obraz przetworzony za pomocą programu SHADE w celu wyrównania jasności tła (b)





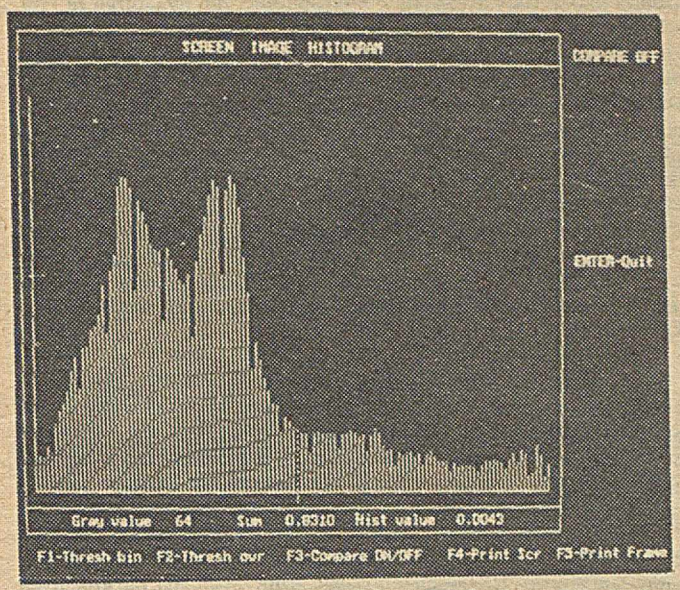
Rys. 16. Rzuty aksometryczne jaskrawości fragmentu obrazu z rys. 15.

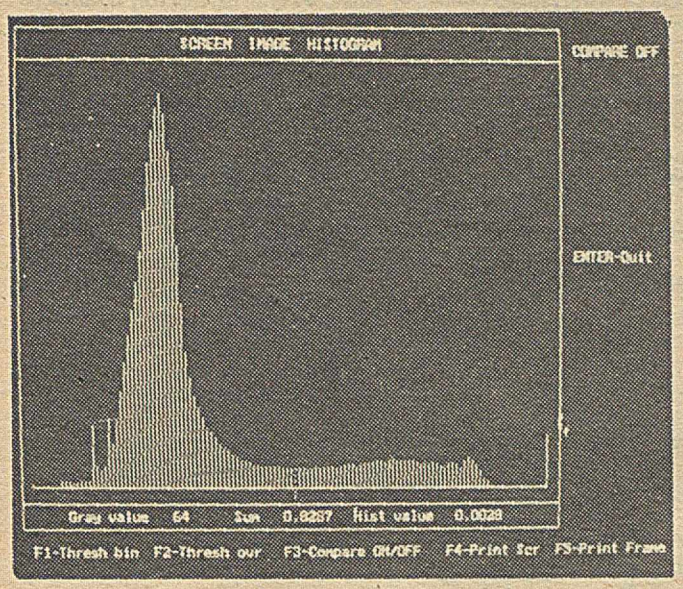
geometryczne i dyskretyzacja obrazu. Wiele pozostaje tu do zrobienia w odniesieniu do półprzewodnikowych elementów światłoczułych, np. matryc obrazowych typu CCD. Przewiduje się, że wyniki tych prac dadzą podstawę do opracowania nowych metod przetwarzania o dużej dokładności, a także umożliwią optymalizację systemów, np. polegającą na opracowaniu reguł doboru parametrów kamery TV do systemu o założonej rozdzielczości.

Wykonane prototypy sterowników rozszerzających i programy komputerowe wdrożono w kilkunastu placówkach, głównie medycznych i akademickich. Zasadniczą korzyścią płynącą z zastosowania systemów przetwarzania obrazów w diagnostyce medycznej jest obiektywizacja i skrócenie czasu badań. Często nie istnieje inna, realistyczna metoda

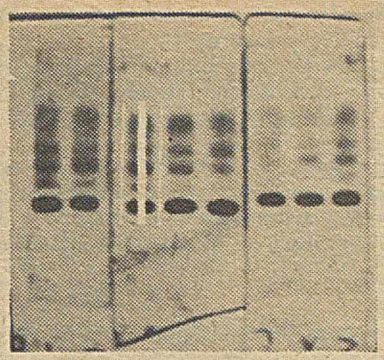
przeprowadzania określonych badań ilościowych, np. w analizie obrazów angiograficznych[1)] lub w biometrii naczyń krwionośnych [8], gdzie jest niezbędna ocena pojemności pojedynczych żył i tętnic o średnicy ułamka milimetra.

[1)] angiografia – rentgenograficzna metoda badania naczyń krwionośnych po wprowadzeniu do nich środka cieniującego.

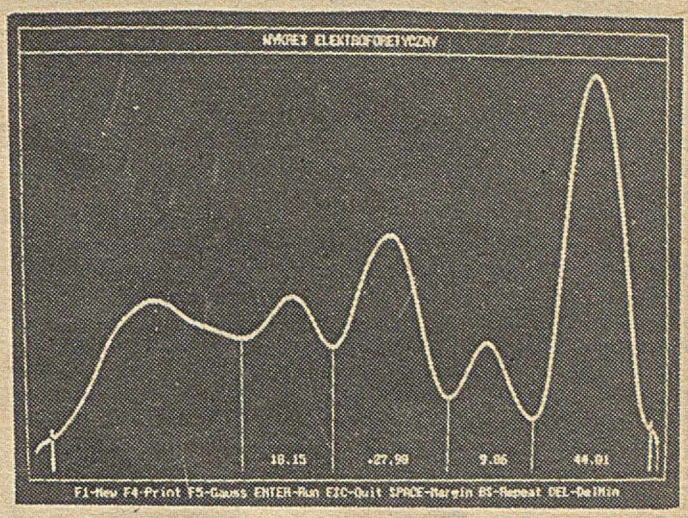




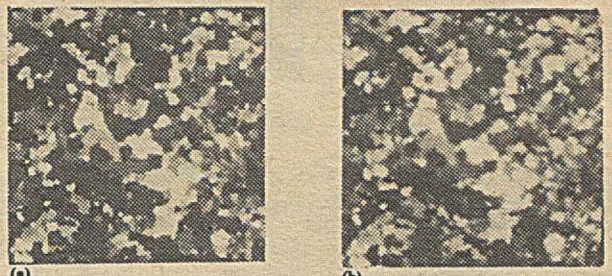
Rys. 17. Histogramy jaskrawości obrazów z rys. 15



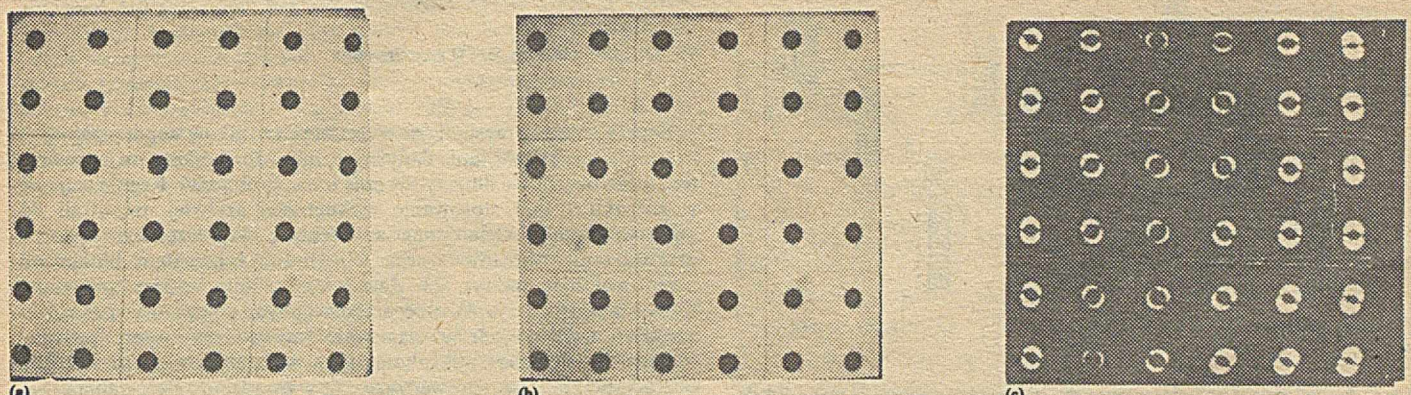
Rys. 18. Obraz preparatu elektroforetycznego z zaznaczoną linią pomiaru gęstości optycznej i liniami korekcyjnymi



Rys. 19. Wygładzony wykres rozkładu gęstości optycznej z obliczonym udziałem procentowym poszczególnych frakcji białka w badanym preparacie



Rys. 20. Obraz powierzchni próbki stopu aluminiowego – szlif metalurgiczny (a); obraz przetworzony za pomocą zestawu filtrów zakresowych (b)



Rys. 21. Obraz wzorcowy, zniekształcony skutkiem nieliniowości odchylania elektronów w kamerze widikonowej (a); obraz po korekcji za pomocą programu WARP (b); obraz ilustrujący różnicę położeń obiektów w obrazach przed korekcją i po korekcji zniekształceń geometrycznych (c)

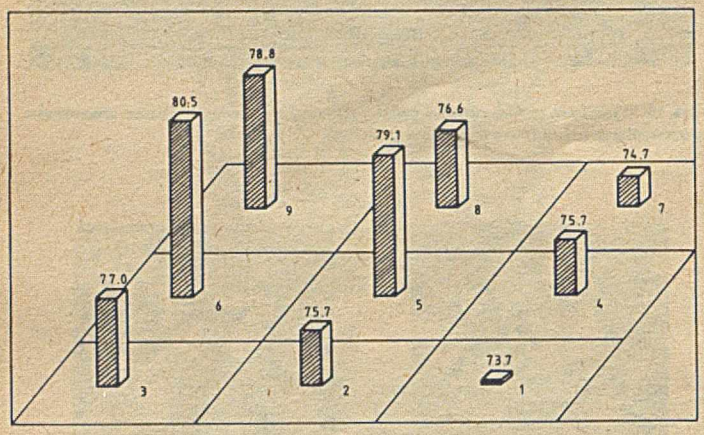
Zastosowanie MSPO w szkolnictwie wyższym daje – przy niewielkim koszcie – wartościową pomoc dydaktyczną. Dotyczy to zwłaszcza kształcenia studentów kierunku Elektroniczna Aparatura Medyczna, którzy otrzymują w ten sposób możliwość symulacji drogiego i niedostępnego sprzętu używanego w medycznej diagnostyce obrazowej.

Inną korzyścią z opracowania MSPO w omawianej tu postaci jest stworzenie projektantom takich systemów możliwości gruntownego przebadania metody przetwarzania obrazów przed jej sprzętową realizacją. Ta forma weryfikacji projektu jest na ogół niedoceniana przez projektantów.

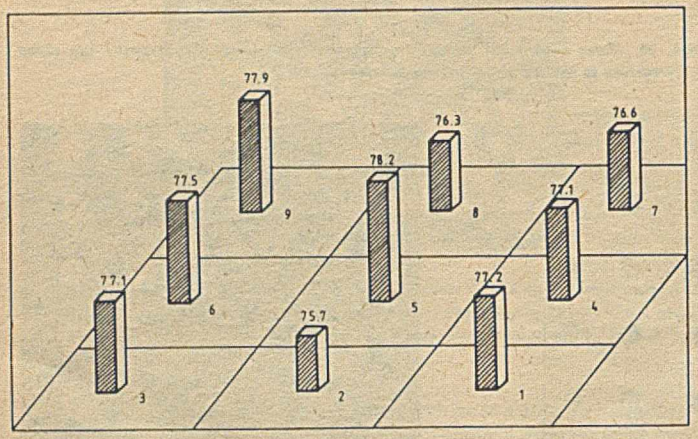
Sterowniki rozszerzające w standardzie IBM PC, do celów przetwarzania obrazów, stanowią istotną część ofert znanych producentów, np. Datacube, Data Translation, Imaging Technology, Imagraph, Matrox Electronic Systems, Truevision, Chorus Data Systems, Univision Technologies i MetraByte Corporation [2]. Firmy te wprowadzają sterowniki wizyjne współpracujące z innymi magistralami, jak Microchanel PS/2, VMEbus, Multibus II, Q-bus oraz Nu-Bus. Można zatem uważać, że koncepcja przekształcania uniwersalnego systemu mikrokomputerowego w system przetwarzania obrazów (przez dołączenie sterowników i oprogramowania) sprawdziła się w praktyce i jest rozwijana. Rozwój ten jest ukierunkowywany na zwiększenie:

- rozdzielczości (liczby piksli obrazu),
- szybkości przetwarzania (skrócenie czasu obliczeń),
- szeroko rozumianej dokładności procesów przetwarzania informacji obrazowej.

Rozdzielczość systemu błędnie bywa uznawana za jedyną miarę jego jakości. Mówi się często, że system przetwarzający obrazy zawierające 1024 × 1024 piksle jest lepszy od systemu, w którym obrazy są tablicami o wymiarze 256 × 256 piksli. W istocie należy podkreślić, że dyskutowanie tego problemu w oderwaniu od potrzeb określonych konkretnym zastosowaniem nie jest, po prostu, możliwe. W wielu zastosowaniach przemysłowych pomiar parametrów technologicznych wyrobu może



(a)

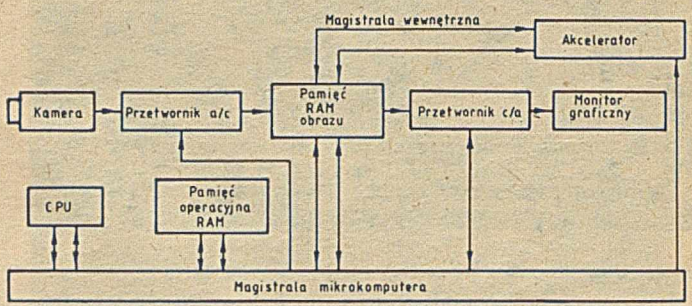


(b)

Rys. 22. Rozkład wartości średniej pola powierzchni modelu koła o promieniu 10 cm w dziesięciu umownych polach obrazu (a); rozkład uzyskany po korekcji zniekształceń geometrycznych programem WARP (b); efektem korekcji jest ponad dwukrotne zmniejszenie rozrzutu pomiarów

być zrealizowany przy umiarkowanej liczbie punktów obrazu, np. w polu o wymiarach 128 × 128 piksli. Dotyczy to, na przykład, rozpoznawania prostych kształtów produkowanych części, pod warunkiem odpowiedniego doboru parametrów układu optycznego kamery. W takich przypadkach używanie sprzętu o dużej rozdzielczości jest niepożądane, zwiększa bowiem koszty urządzeń (pamięć obrazu) i wydłuża czas obliczeń ponad rzeczywiste potrzeby. Z drugiej strony, systemy wspomagania medycznej diagnostyki rentgenowskiej wymagają rozdzielczości 1024 × 1024 piksli do zróżnicowania bliskich, np. odległych od siebie o 0,5 mm szczegółów zdjęcia RTG. Podobnie, rezultaty badań statystycznych w mikrobiologii lub metalurgii są tym pewniejsze, im większa jest liczba obiektów analizowanych w polu pojedynczego obrazu. Przy ustalonych wymiarach bezwzględnych obiektów i odległościach między nimi, zwiększenie liczby obiektów w obrazie wymaga jednoczesnego zmniejszenia odstępów próbkowania i zwiększania pola widzenia, a zatem większej rozdzielczości systemu. Oryginalne rozwiązanie tego problemu przedstawiono w systemie **PIPS [18]**, w którym liczba punktów obrazu cyfrowego (rozdzielczość) jest parametrem systemu, sterowanym programowo. Dążenie do uzyskania dużej rozdzielczości można uznać obecnie za trwałą tendencję w rozwoju MSPO, przy czym jednym z czynników hamujących jest w tym przypadku słabe rozpowszechnienie odpowiednich kamer. Typowe kamery TV umożliwiają zapamiętanie obrazów 512 × 512 lub 720 × 560 piksli.

Większość funkcji przetwarzania obrazów można podzielić na operacje punktowe i lokalne. W wyniku działania operacji pierwszego rodzaju jaskrawość każdego punktu w obrazie wynikowym jest określoną funkcją jaskrawości punktu o tych samych współrzędnych w obrazie źródłowym. Przy operacjach lokalnych jaskrawość każdego punktu w obrazie wynikowym jest funkcją jaskrawości punktów z pewnego sąsiedztwa odpowiedniego punktu w obrazie źródłowym. W obu przypadkach, wykonanie procedury polega na wielokrotnym wykonywaniu elementarnych operacji punktowych lub lokalnych, dla kolejnych piksli przetwarzanego obrazu. W związku z tym, istnieją trzy zasadnicze metody przyspieszenia procedur przetwarzania obrazów. Pierwsza polega na optymalizacji procedur elementarnych, powtarzanych wielokrotnie. W systemach z jednym procesorem regułą jest kodowanie algorytmów takich procedur w języku maszynowym. Drugie rozwiązanie opiera się na stosowaniu tzw. akceleratorów [18], to jest specjalizowanych układów arytmetyczno-logicznych lub procesorów sygnałowych, realizujących określoną funkcję na pojedynczym obrazie w czasie rzeczywistym, czyli w czasie porównywalnym do czasu trwania jednego obrazu TV (40 ms). Według trzeciej, najdroższej metody, przyspieszenie pracy systemu uzyskuje się przez zastosowanie przetwarzania potokowego lub równoległego.



Rys. 23. Schemat blokowy MSPO z akceleratorem

W przypadku korzystania ze sprzętowych środków wspomagających obliczenia, czas przesyłania danych z pamięci do akceleratora i z powrotem może okazać się dłuższy niż czas trwania obliczeń. Regułą staje się w związku z tym stosowanie wewnętrznej szybkiej magistrali do mikrokomputera. Ostatecznie architekturę współczesnego systemu przetwarzania obrazów można zilustrować schematem blokowym przedstawionym na rys. 23. Zasadnicza różnica między systemami tradycyjnymi (por. rys. 4) a obecnie rozwijanymi polega na wyposażeniu urządzeń tradycyjnych w magistralę wewnętrzną i specjalizowany akcelerator. Procesor mikrokomputera wykonuje w takim systemie operacje pomocnicze, sprowadzające się w zasadzie do sterowania pracą urządzenia. Ważną zaletą nowego rozwiązania jest jego modułowość. Zmiana klasy przetwarzanych obrazów lub metod przetwarzania

wymaga zmiany akceleratora albo odpowiedniej zmiany programu w przypadku uniwersalnych akceleratorów programowanych.

Omawiane systemy służą do przetwarzania obrazów monochromatycznych. Można się spodziewać, że w niedalekiej przyszłości uwaga konstruktorów i programistów będzie poświęcona w znacznej części opracowaniu efektywnych metod przetwarzania obrazów kolorowych.

LITERATURA

[1] Abramczuk T.: Portable Image Processing System. Brief Description and Users Manual. The Royal Institute of Technology. Instrumentation Laboratory, Stockholm, 1989 (dzięki uprzejmości Prof. R. B. Johanssona)

[2] Curran L.: Imaging Boards Focus on High Resolution. Electronic Design. pp. 43–49, September 1989

[3] Gielecki J., Materka A.: Zastosowanie systemu cyfrowej analizy obrazów w badaniach biometrycznych układu naczyń krwionośnych. Krajowa Konferencja Naukowa „Komputery w medycynie". pp. 99–104, Łódź 1989

[4] Gocławski J., Materka A., Pałuba M., Sitek A., Stetkiewicz J., Strumiłło P., Zatorski W.: Mikrokomputerowy system do przetwarzania obrazów medycznych dla komputerów IBM PC. Archiwum Informatyki Teoretycznej i Stosowanej. No 1, 1989

[5] Gościcka D., Gielecki J., Materka A.: Zastosowanie mikrokomputerowego systemu analizy obrazu do pomiaru parametrów naczyń koła tętniczego mózgu. Folia Morphologica, No. 1, 1989

[6] Haralick R.M., Zhuang X., Lin C., Lee J.S.J.: The Digital Morphological Sampling Theorem. IEEE Trans. Acoust. Speach Signal Proc., Vol. 37, No.12, pp. 2067–2090, December 1989

[7] Małafiej E., Materka A., Denys A.: Komputerowe metody wyznaczania aktywności antybiotyków w badaniach in vitro. Krajowa Konferencja Naukowa „Komputery w medycynie", pp. 185–190, Łódź 1989

[8] Materka A.: Digital Image Analysis in Biometric Research of Blood Vessel Systems. „Komputerowe Przetwarzanie Obrazów Biomedycznych". Seminarium Międzynarodowego Centrum Biocybernetyki, Mądralin 1989

[9] Materka A.: Estymacja pojemności naczyń krwionośnych metodą cyfrowej analizy obrazów. XII KTOiUE. pp. 301–305, Myczkowce 1989

[10] Materka A., Gielecki J.: Application of Digital Image Analysis System to Blood Vessels Geometric Parameters Estimation. 12th Int. Congress on X-Ray Optics and Microanalysis, pp. 630–633, Kraków 1989

[11] Materka A., Gielecki J.: Obliczanie parametrów geometrycznych naczyń krwionośnych za pomocą mikrokomputerowego systemu przetwarzania obrazów. Archiwum Informatyki Teoretycznej i Stosowanej. No 1, 1989

[12] Materka A., Górnisiewicz W., Kukuła M.: Skalowanie i korekcja zniekształceń geometrycznych w systemie cyfrowego przetwarzania obrazów z kamerą widikonową (artykuł złożony w redakcji „Rozpraw Elektrotechnicznych")

[13] Materka A., Strumiłło P.: Metoda pomiaru pola powierzchni obiektów w obrazach silnie zaszumionych. XI KTOiUE, pp. 136–141, Rytro 1988

[14] Materka A., Strumiłło P.: Metoda wstępnego przetwarzania obrazów kolonii bakteryjnych. XI KTOiUE, pp. 142–146, Rytro 1988

[15] Materka A., Strumiłło P.: Wpływ wybranych zniekształceń sygnału wizyjnego na błąd pomiaru pola powierzchni figur płaskich w cyfrowych systemach analizy obrazów. XI KTOiUE, pp. 295–300, Myczkowice 1989

[16] Materka A., Strumiłło P.: Wybrane problemy pomiaru pola powierzchni figur płaskich w cyfrowych systemach analizy obrazów. Archiwum Informatyki Teoretycznej i Stosowanej. No 1, 1989

[17] Mikrokomputerowy system do przetwarzania obrazów dla komputerów klasy IBM PC/XT. Podręcznik użytkownika. CPBR 8.6, etap II, Łódź 1989

[18] Opracowanie metody estymacji średnic i liczby kropel w obrazach halograficznych strugi rozpylonego paliwa za pomocą mikrokomputerowego systemu obrazowego. Sprawozdanie z pracy naukowo-badawczej, na zlecenie Instytutu Pojazdów PŁ. Łódź 1988

[19] Pennley W.: Images. BYTE, pp. 248–256, Dec. 1989.

IF Informatyk THEN Kto? ELSE Programista

dokończenie z III strony okładki

przypadkach trudno jest mu zrozumieć istotę algorytmów obliczeń fizycznych, chemicznych, mechanicznych czy przetwarzania informacji.

Tylko przy takim układzie współdziałania informatyka z użytkownikiem możliwe jest prawidłowe eksploatowanie istniejących obecnie narzędzi informatyki oraz dalszy ich rozwój w kierunku lepszego przystosowania do potrzeb innych użytkowników. Trzeba bowiem zdawać sobie sprawę z tego, że w coraz mniejszym stopniu użytkownik komputerów będzie korzystał bezpośrednio z pośrednictwa specjalistów z ośrodka komputerowego. Coraz większa będzie też grupa użytkowników nie mająca i nawet nie chcąca mieć jakiegokolwiek pojęcia o funkcjonowaniu sprzętu komputerowego i zasadach konstrukcji programów w dotychczasowych językach programowania. Jednocześnie rozwój informatyki gwałtownie powiększa moc obliczeniową komputerów, przez konstruowanie coraz szybszych procesorów oraz tworzenie rozproszonych wieloprocesorowych systemów przetwarzania danych. Pociąga to za sobą znaczny stopień skomplikowania tych rozwiązań, uniemożliwiając chałupniczą eksploatację. Powracając do przykładu z motoryzacji, silnik malucha mógł rozłożyć i złożyć na podwórku średnio zaawansowany mechanik-samouk. Silnika nowoczesnego samochodu już nikt nie może w taki sposób naprawiać.

Pozostała jeszcze grupa informatyków zajmująca się teorią informatyki. Definicja zakresu ich działania nie wymaga sprecyzowania. Bronią się oni sami tworząc coraz bardziej zmatematyzowane teorie z zakresu specyfikacji i dowodzenia poprawności programów sekwencyjnych i współbieżnych, teorii obliczeń i algorytmów, a także podstaw teorii baz danych, systemów ekspertowych itp. Jednak i tutaj należy wskazać na rolę informatyków-praktyków, którzy powinni – gdy prace teoretyczne sugerują możliwość praktycznego rozwiązania zagadnienia – przystąpić do jego wdrożenia, a następnie spopularyzowania wersji eksploatacyjnej wśród potencjalnych użytkowników.

Informatykiem nie jest więc ten, który złożył mikrokomputer IBM PC/486 z części zakupionych na Tajwanie. Informatykiem nie jest też ten, kto potrafi jedynie programować w jakimś języku, a nawet napisał kilkanaście bardzo dużych i złożonych programów. Jeżeli programy te dotyczyły zagadnień związanych z jego zawodem, można powiedzieć, że stał się on lepszym specjalistą w swojej dziedzinie, wykorzystującym nowoczesną technikę. Informatykiem nie jest też stały czytelnik pisma BYTE, znający na wyrywki najnowsze wersje produktów firm Microsoft i Borland. Ażeby być informatykiem potrzebna jest rzetelna wiedza podstawowa, aktualna nie tylko teraz, ale również stanowiąca podstawę do jej przyszłego poszerzenia.

Informatyk musi być też odpowiedzialny za to, co robi, co proponuje wykorzystać, czego naucza. Odpowiedzialność ta stale się powiększa, gdyż coraz więcej dziedzin życia zaczyna zależeć od komputerów. Obliczenia wykonywane z pomocą komputerów, nadzór komputerowy nad procesami technologicznymi, aparatura kontrolno-pomiarowa i medyczna, gromadzenie i przetwarzanie informacji o społeczeństwie i jego jednostkach mogą prowadzić w przypadku błędnego działania systemów liczących do nieobliczalnych skutków.

Podsumowaniem tego tematu niech będzie cytat z felietonu „Od Prezesa" Prof. A. Bliklego, zamieszczony w numerze 1, 1990 Biuletynu PTI:.

„Każdy ma prawo zaaplikować sobie lekarstwo z domowej apteczki lub zbudować szopę w ogrodzie. Nikt jednak nie poddaje w wątpliwość zasady, że ani leczyć, ani budować nie wolno bez odpowiednich uprawnień. A uprawnienia takie uzyskuje się kończąc wyższą uczelnię, a nie kilkutygodniowy czy kilkumiesięczny kurs.

Uważam, że w świadomości społecznej powinno utrwalić się zrozumienie faktu, że informatyka jest zawodem równie trudnym i odpowiedzialnym, jak medycyna czy inżynieria budowlana, i że jedyną drogą do jej opanowania są studia zawodowe. Ruch amatorski należy oczywiście popierać również w informatyce, trzeba jednak pamiętać, że amatorska informatyka może być używana jedynie w amatorskich zastosowaniach. Natomiast zastosowania profesjonalne wymagają profesjonalnej informatyki".

WACŁAW ISZKOWSKI

P.S. Zbitki poglądów i spostrzeżeń tutaj prezentowanych nie są oficjalnym stanowiskiem Redakcji ani żadnej innej organizacji. Ich prezentacja ma służyć pobudzeniu dyskusji na dany temat nawet przez wzburzenie moich ewentualnych adwersarzy.

Informujemy Czytelników,

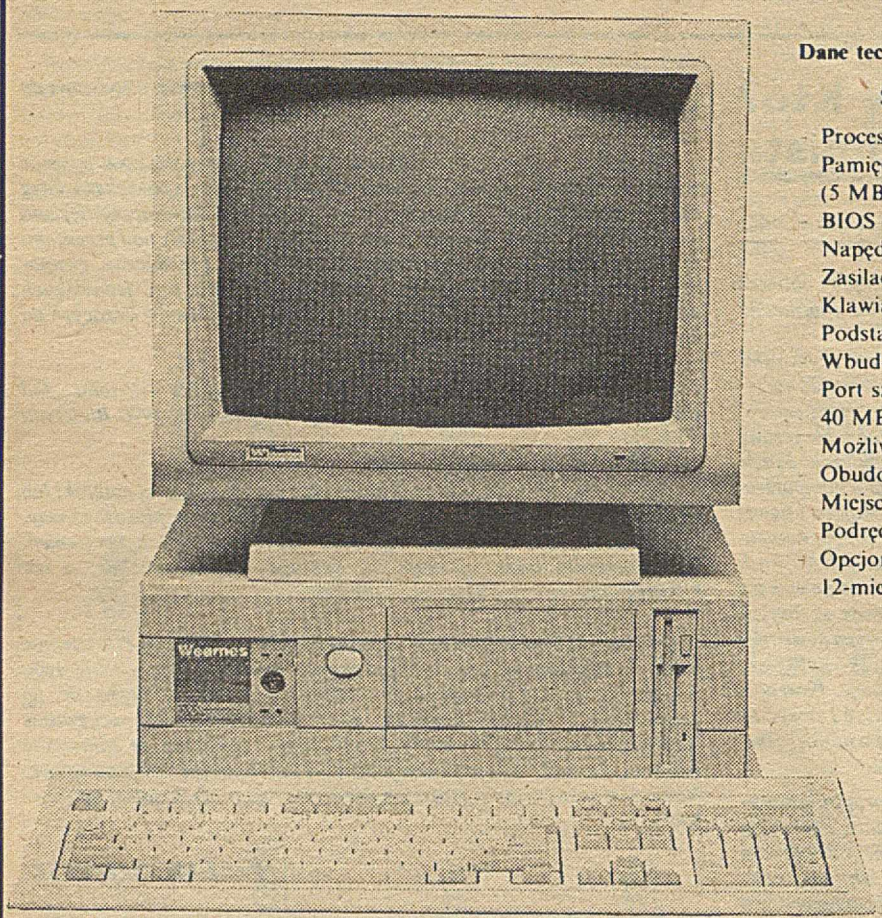
że blankiety do zamawiania prenumeraty dla odbiorców zbiorowych (instytucji) stałym prenumeratorom będą dostarczane przez Zakład Kolportażu. Nowi prenumeratorzy proszeni są o zgłaszanie się po blankiety do redakcji (tel. 39-14-34) lub do Zakładu Kolportażu (tel. 40-30-86 lub 40-35-89).

KOMPUTER
Z WBUDOWANĄ PRZYSZŁOŚCIĄ

Komputer zaprojektowany przez czołowego światowego producenta mikrokomputerów – firmę ADVANCED LOGIC RESEARCH i wyprodukowany przez firmę WEARNES TECHNOLOGY.

Komputer, którego możliwości i cena oszałamiają konkurencję.

Komputer, który nie zestarzeje się, dzięki możliwości rozbudowy do architektury 386SX i 486.



Dane techniczne:

SERIA WEARNES BOLDLINE „M"

- Procesor 80286 12.5 Mhz
- Pamięć 1 MB RAM, możliwość rozbudowy do 16 MB (5 MB na płycie głównej)
- BIOS Phoenix
- Napęd dyskietek 3,5" 1,44 MB
- Zasilacz 110 Watt
- Klawiatura 101 klawiszy
- Podstawka dla koprocesora matematycznego 80287
- Wbudowany sterownik dyskowy z przeplotem 1:1
- Port szeregowy i równoległy
- 40 MB dysk sztywny
- Możliwość korzystania z EMS 4.0
- Obudowa typu „compact"
- Miejsce na dwa napędy 5,25" o wysokości 1/2
- Podręcznik i dyskietka z programem konfiguracyjnym
- Opcjonalna rozbudowa do 386SX i 486
- 12-miesięczna gwarancja

Komputer z wbudowaną przyszłością!

BOLDLINE COMPUTERS
GRUPA MICOMP-TECH
Biuro Informacji Techniczno-Handlowej
ul. Astrów 7, 40–045 Katowice
telefon i telefaks: 518-628
teleks: 315687 COMP PL

DYSTRYBUTORZY:
PTH „TECHMEX"
43–300 Bielsko-Biała
ul. Curie-Skłodowskiej 13
tel.: 42-198, 47-555, telefaks: 47-624,
teleks: 35325

EO/1123/89

WIESŁAW LUCJAN NOWIŃSKI
Instytut Podstaw Informatyki PAN
Warszawa

Problemy współbieżnej rekonstrukcji obrazów

Rekonstrukcja obrazów z rzutów została przedstawiona w ubiegłorocznych numerach (3, 4 oraz 5) INFORMATYKI [6, 7, 8]. Przypomnijmy, że jest to technika umożliwiająca zobrazowywanie wewnętrznej struktury badanych obiektów na podstawie odpowiedniej informacji pomiarowej. Rekonstrukcja obrazów jest stosowana w wielu dziedzinach, jak np. diagnostyka medyczna, zastosowania przemysłowe, radioastronomia, czy też mikroskopia elektronowa. Jednakże najpowszechniejszym zastosowaniem rekonstrukcji obrazów jest tomografia komputerowa, przedstawiona w nr 7, 1989 INFORMATYKI [14].

Pierwsze tomografy komputerowe na zebranie danych pomiarowych i zrekonstruowanie obrazu potrzebowały kilka minut. W zastosowaniach medycznych powodowało to degradację zrekonstruowanego obrazu wynikającą z ruchu pacjenta w czasie zbierania danych pomiarowych. Obecnie stosowane tomografy komputerowe nie mają już tej wady, gdyż zbierają dane pomiarowe i rekonstruują obraz w czasie kilku sekund. Jednakże w wielu zastosowaniach badany obiekt wykonuje szybkie ruchy lub podlega deformacjom, w których to sytuacjach klasyczna tomografia komputerowa daje obrazy rozmyte. W tych przypadkach niezbędne jest stosowanie tzw. szybkiej tomografii komputerowej. Dla potrzeb szybkiej tomografii komputerowej skonstruowano urządzenia umożliwiające zbieranie danych pomiarowych w czasie mili- lub nawet mikrosekund. Przykładem takich urządzeń są układy DSR (ang. *Dynamic Spatial Reconstructor*) [13], czy też CVCT (ang. *Cardiovascular Computed Tomograph*) [12]. Układ DSR może generować w ciągu jednej sekundy dane pomiarowe aż do 14 400 rekonstrukcji. Wymagania medyczne określają czas badania serca na cztery sekundy. Stąd też całkowita liczba wykonywanych rekonstrukcji może dochodzić do 57 600. Przetworzenie tak dużej liczby danych w rozsądnym czasie wymaga przetwarzania współbieżnego. W zastosowaniach przemysłowych wymagania dotyczące czasu zbierania danych mogą być większe niż w zastosowaniach medycznych, jak np. w śledzeniu na bieżąco procesów szybkozmiennych. Wówczas współbieżną rekonstrukcję obrazów wykorzystuje się nie tylko ze względu na konieczność przetworzenia dużej liczby danych pomiarowych, lecz przede wszystkim na konieczność wykonywania rekonstrukcji w czasie rzeczywistym.

Przetwarzanie współbieżne obejmuje trzy dziedziny badań (grupy problemów): języki programowania współbieżnego, współbieżne algorytmy oraz współbieżne architektury. Rozpatrując problematykę z zakresu współbieżnej rekonstrukcji obrazów omówimy te trzy dziedziny.

Języki programowania współbieżnego

Współbieżność przetwarzania istnieje na różnych poziomach: algorytmu, języka programowania oraz sprzętu użytego do implementacji algorytmu. Weźmy np. algorytm rekonstrukcji, w którym rzuty mogą być przetwarzane współbieżnie. Wówczas stopień współbieżności przetwarzania na poziomie algorytmu równa się liczbie rzutów. Jeśli jednak ten algorytm zapiszemy w sekwencyjnym języku programowania, to otrzymamy stopień współbieżności przetwarzania równy jeden. Taki sam rezultat będzie, jeśli rozpatrywany algorytm zapiszemy w języku programowania współbieżnego, lecz wykonamy go na systemie jednoprocesorowym.

Jednym ze sposobów opisu współbieżności na poziomie algorytmu jest wykorzystanie języków programowania współbieżnego. Języki programowania współbieżnego dzieli się zwykle na trzy grupy:
- **asynchroniczne języki programowania współbieżnego**, takie jak np. Modula-2, Pascal Plus, Ada, Occam. W językach tych w czasie wykonywania programu procesy komunikują się w celu przekazywania informacji lub synchronizacji swoich działań,
- **synchroniczne języki programowania współbieżnego**, w których (wykorzystując regularności struktur danych) procesy składowe są wykonywane synchronicznie. Przykładem takich języków są Actus lub Fortran dla superkomputerów wektorowych (jak Cray-1 lub CDC Cyber 205) lub tablicowych (jak Illiac IV lub ICL Distributed Array Processor),
- **języki programowania tzw. „data flow"** (np. VAL), w których kilka instrukcji może być wykonywanych współbieżnie pod warunkiem, iż mają one dostęp do odpowiednich danych.

Do opisu współbieżnej rekonstrukcji obrazów wykorzystano język Occam. Stąd też przed sformułowaniem współbieżnych algorytmów rekonstrukcji obrazów przedstawimy pokrótce podstawowe pojęcia tego języka (został on szczegółowo opisany w INFORMATYCE w numerach 7 oraz 8 z 1988 r.)

W języku Occam występują trzy następujące procesy pierwotne: przypisania, wejściowy oraz wyjściowy. Procesy pierwotne stanowią składniki, z których za pomocą konstruktorów można budować procesy złożone. Podstawowe konstruktory to konstruktor sekwencyjny *SEQ* i konstruktor współbieżny *PAR*. Konstruktor sekwencyjny SEQ powoduje, iż procesy składowe wykonywane są jeden po drugim.

(1) *SEQ*
{*proces*}

gdzie zapis {*proces*} oznacza listę procesów.
Konstruktor współbieżny *PAR* tworzy proces złożony, którego procesy składowe wykonywane są współbieżnie

(2) *PAR*
{*proces*}

W celu stworzenia możliwości powielenia procesu wprowadzono konstruktor repetycyjny *REP*. Konstruktor ten może być używany w połączeniu z innymi konstruktorami, jak np. *SEQ* lub *PAR*. Ogólna postać konstruktora repetycyjnego jest następująca:

(3) *REP indeks = wartość początkowa indeksu FOR liczba powtórzeń*

gdzie *REP* jest konstruktorem, jak np. *SEQ or PAR*.
Repetycyjny konstruktor *SEQ* jest równoważny pętli (którą można również utworzyć np. za pomocą konstruktora *WHILE*), natomiast repetycyjny konstruktor *PAR* tworzy tablicę identycznych procesów współbieżnych, do których można się odwoływać za pomocą indeksu.

Współbieżne algorytmy rekonstrukcji obrazów

Formy współbieżności przetwarzania w rekonstrukcji obrazów określone są przez sposoby dekompozycji procesów złożonych na procesy składowe. Forma współbieżności jest scharakteryzowana przez:
- funkcje wykonywane przez procesy składowe,
- dane, na których operują procesy składowe,
- obiekty, do których są przypisane procesy składowe.

W algorytmach rekonstrukcji obrazów istnieje wiele form współbieżności przetwarzania. Formy te zostały wyodrębnione i zdefiniowane za pomocą języka Occam. Podstawowe z nich to współbieżność:
- przetwarzania piksli (lub woksli w rekonstrukcji trójwymiarowej),

- przetwarzania rzutów,
- przetwarzania promieni,
- operacji.

Ponadto do rekonstrukcji trójwymiarowej wprowadzono współbieżność przetwarzania plastrów.

Współbieżność przetwarzania piksli, rzutów, promieni i plastrów zdefiniowano za pomocą konstruktora repetycyjnego z konstruktorem PAR [8]; wynika to z faktu, iż procesy składowe tworzą tablice identycznych procesów. Współbieżność operacji zdefiniowano za pomocą konstruktora współbieżnego *PAR* [7]. Formalne definicje tych form współbieżności przetwarzania można znaleźć w pracy [9].

Współbieżne algorytmy rekonstrukcji obrazów można sformułować wykorzystując powyższe formy oraz konstruktory języka Occam, jak *SEQ, PAR, REP, WHILE.*

Rozważmy transformacyjne metody rekonstrukcji obrazów [7]. Należą do nich metody: splotu, Fouriera i filtrowanego rzutowania zwrotnego.

Metoda splotu jest najpopularniejszą metodą rekonstrukcji obrazów. Rekonstruuje ona obraz w dwóch krokach: filtracja rzutów i rzutowanie zwrotne. Pierwszy krok można wykonać wykorzystując współbieżność przetwarzania rzutów, w drugim kroku natomiast można wykorzystać współbieżność przetwarzania piksli. Stąd też współbieżny algorytm dla metody splotu jest następujący:

```
PROC metoda.splotu (VAL INT M, J)
INT m, j:
SEQ
  PAR m = 0 FOR M          — M – liczba rzutów
  filtracja.rzutów (m)
  PAR j = 0 FOR J          — J – liczba piksli
  rzutowanie.zwrotne (j)
  :
```

Dla metody splotu można sformułować dwa dalsze algorytmy współbieżne wykorzystując fakt, iż rzutowanie zwrotne można wykonywać współbieżnie wykorzystując współbieżność przetwarzania rzutów i współbieżność operacji.

Metoda rekonstrukcji Fouriera może być zaimplementowana współbieżnie wykorzystując współbieżność operacji. Istnieje ona zarówno w transformacie Fouriera, jak i w interpolacji biliniowej.

Metodę filtrowanego rzutowania zwrotnego można zaimplementować współbieżnie wykorzystując zarówno współbieżność istniejącą w transformacie Fouriera, jak i formy współbieżności przetwarzania występujące w operacji rzutowania zwrotnego.

Rozważmy metody rozwinięcia w szereg [8]. Metody te dzielą się na metody iteracyjne (algebraiczne i symultacyjne) i nieiteracyjne. Algorytmy algebraiczne (tzn. algorytmy typu ART) przetwarzają w jednym kroku tylko jedno równanie, stąd też przetwarzanie równań nie może być wykonywane współbieżnie. Jednakże przetwarzając każde równanie można wykorzystać współbieżność operacji, jak np. współbieżne obliczanie iloczynów skalarnych w metodzie relaksacyjnej.

W symultacyjnych algorytmach rekonstrukcji (tzn. algorytmach typu SIRT) istnieją różne formy współbieżności przetwarzania. Wykorzystując współbieżność przetwarzania rzutów otrzymujemy:

```
PROC algorytmy.sirt.współbieżność.rzutów (VAL INT M, N)
  WHILE NOT warunek.zbieżności
    PAR m = 0 FOR M
      przetwarzaj.jeden.rzut (N, m)
:
```

gdzie procedura *przetwarzaj.jeden.rzut (N, m)* przetwarza $2N+1$ równań odpowiadających m-temu rzutowi. Przetwarzanie wszystkich równań wykonywane jest iteracyjnie, aż do spełnienia warunku zbieżności. Zakładając, że wszystkie równania mogą być przetwarzane współbieżnie, można wykorzystać współbieżność przetwarzania promieni:

```
PROC algorytmy.sirt.współbieżność. promieni (VAL INT I)
  WHILE NOT warunek. zbieżności
    PAR i = 0 FOR I          — I – liczba promieni
      przetwarzaj.jedno.równanie (i)
:
```

W metodzie nieiteracyjnej z minimalnym błędem dokonuje się dekompozycji układu równań na $2V+1$ podukładów. Wówczas można wykorzystać współbieżność operacji:

```
PROC metoda.nieiteracyjna.min.błąd (VAL INT V)
  INT v:
  PAR v = – V FOR (2*V+1)
    podukład.równań (v)
:
```

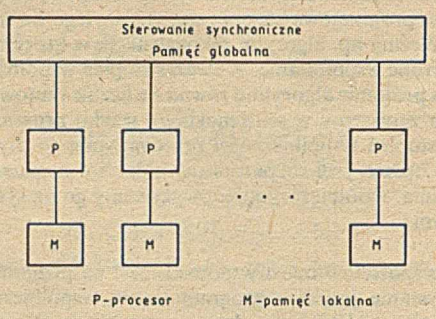
Analizując własności współbieżności danego problemu należy rozpatrywać dodatkowe czynniki, takie jak:

- warunki istnienia form współbieżności przetwarzania,
- osiągnięte w wyniku współbieżności przetwarzania przyspieszenie prędkości obliczeń (ang. *speedup*),
- wykorzystanie pamięci (wielkość pamięci, liczba odwołań do pamięci, równoczesny dostęp do pamięci, pamięci współbieżne),
- komunikacja między procesami,
- ziarnistość procesów,
- sterowanie współbieżnością,
- wykorzystanie procesorów i ich obciążenie (ang. *workload*),
- koszt rozwiązania.

Architektury do współbieżnej rekonstrukcji obrazów

Rozważmy propozycje architektur odpowiednich do rekonstrukcji obrazów. Należałoby przypuszczać, iż do współbieżnej rekonstrukcji obrazów można wykorzystać architektury odpowiednie do współbieżnego przetwarzania obrazów. Architekturą odpowiednią do współbieżnego przetwarzania obrazów jest architektura typu SIMD (ang *Single Instruction Multiple Data*) z jednobitowymi elementami przetwarzającymi. Jednakże charakter przetwarzania obrazów jest różny od charakteru rekonstrukcji obrazów. Jedna z różnic tkwi w występowaniu różnych form współbieżności i przetwarzania. Zauważmy, iż w formach współbieżności przetwarzania w rekonstrukcji obrazów nie występuje współbieżność przetwarzania sąsiedztwa (ang. *neighbourhood parallelism*) charakterystyczna dla przetwarzania obrazów. Brak tej formy współbieżności eliminuje konieczność istnienia połączeń między sąsiednimi elementami przetwarzającymi. W konsekwencji współbieżne systemy komputerowe do przetwarzania obrazów mają sieci połączeń niedopasowane do rekonstrukcji obrazów.

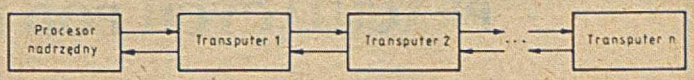
Algorytmy rekonstrukcji obrazów wymagają dostępu do pamięci globalnej zawierającej dane pomiarowe, rekonstruowany obraz i, ewentualnie, wstępnie przetworzone dane. To wymaganie jest charakterystyczne dla architektury typu MIMD (ang. *Multiple Instruction Multiple Data*). Z drugiej jednak strony procesy składowe dla współbieżności przetwarzania piksli, rzutów, promieni i plastrów (jak również w wielu przypadkach współbieżności operacji) tworzą tablice identycznych procesów współbieżnych. Z tego też powodu algorytmy wykorzystujące te formy równoległości są synchroniczne i mają scentralizowane sterowanie. Te cechy są charakterystyczne dla architektury typu SIMD. Formalnym modelem współbieżnego przetwarzania łączącym zalety obu architektur jest PRAM (ang. *Parallel Random Access Machine*). PRAM zakłada możliwym użycie wielu identycznych procesorów pracujących synchronicznie w trybie SIMD i wykorzystujących globalną (współdzieloną) pamięć (rys. 1).



Rys. 1 Architektura typu PRAM

Z tego też powodu architekturę typu PRAM przyjęto jako architekturę charakterystyczną dla rekonstrukcji obrazów [10]. W praktycznych realizacjach jednakże liczba procesorów dla architektury typu PRAM jest ograniczona. Przy obecnie stosowanej technologii nie zbudowano jeszcze systemu o architekturze PRAM zawierającego ponad 100 procesorów. Z drugiej jednak strony stopień współbieżności w rekonstrukcji obrazów może być znacznie większy. Przykładowo, dla obrazu składającego się z 256 × 256 pikseli stopień ten wynosi ponad 65000 dla współbieżności przetwarzania pikseli, natomiast stosując wiązkę promieni złożoną z 500 promieni przy 500 rzutach stopień współbieżności wynosi 250000 dla współbieżności promieni. Stąd też dla konkretnych metod (lub grup metod) należy poszukiwać architektur, których realizacja umożliwi w pełni wykorzystanie współbieżności przetwarzania istniejącej w algorytmach rekonstrukcji obrazów. Przykładem takich architektur są dwie architektury typu SIMD dla metody splotu [9]. W rozwiązaniach tych uzyskano bezkonfliktowy dostęp do pamięci dzięki odpowiednio zaprojektowanej sieci przełączającej. Jedno z rozwiązań wykorzystuje współbieżność przetwarzania pikseli. W tym przypadku rekonstruowany obraz znajduje się w pamięciach lokalnych, natomiast dane pomiarowe są w pamięci globalnej (składającej się z bloków i dostępnej współbieżnie). W drugim rozwiązaniu, wykorzystującym współbieżność przetwarzania rzutów, w pamięciach lokalnych przechowywane są dane pomiarowe, natomiast rekonstruowany obraz znajduje się w pamięci globalnej. Jeszcze inną architekturę typu SIMD przedstawiono w pracy [1]; oparta jest ona na współbieżności operacji przy wykonywaniu rzutowania zwrotnego i wykorzystuje procesory do współbieżnego obliczania iloczynu skalarnego.

Powyższe rozwiązania wykorzystują algorytmy synchroniczne. Jednakże ze względu na asynchronizm występujący w rekonstrukcji obrazów, stosuje się również architektury typu MIMD. W pracy [3] do współbieżnej rekonstrukcji obrazów użyto dwuwymiarową tablicę transputerów. Rozwiązanie to wykorzystuje współbieżność przetwarzania pikseli, aczkolwiek dane przesyłane są wzdłuż promieni. Daje to w wyniku małe wykorzystanie transputerów (od 10% do 35%) i nierównomierne ich obciążenie. Lepsze wykorzystanie transputerów daje liniowa tablica transputerów zaproponowana w pracy [11] do współbieżnej implementacji metod symultacyjnych (rys. 2).



Rys. 2 Liniowa tablica transputerów

Rozwiązanie to wykorzystuje współbieżność przetwarzania rzutów lub współbieżność przetwarzania promieni. W tym przypadku pamięć globalna jest w komputerze nadrzędnym, natomiast transputery wykorzystują współbieżność komunikacji z przetwarzaniem.

Ponadto, algorytmy rekonstrukcji obrazów wykorzystujące transformatę Fouriera lub operację splatania mogą wykorzystywać architektury odpowiednie do tych operacji, jak SIMD [2], MIMD [5] lub struktury systoliczne [4].

LITERATURA

[1] Gilbert B.K. i in.: Rapid execution of fan beam image reconstruction algorithms using efficient computational techniques and special-purpose processors. IEEE Trans. Biomed. Eng., Vol. BME-28, No. 2, 1981
[2] Jesshope C.R.: The implementation of fast radix 2 transforms on array processors. IEEE Trans. Comput., Vol C-29, No. 1, 1980
[3] Kingswood N. i in.: Image reconstruction using the transputer. IEE Proceedings, Vol. 133, Pt. E., No. 3, 1986
[4] Kung H.T.: The structure of parallel algorithms. Advances in Comps., 19, 1980

dokończenie na s. 26

INTERSOFT Sp. z o.o.

Nasze Biuro Handlu Zagranicznego poleca swoje usługi w zakresie eksportu-importu każdego towaru. Prowadzi sprzedaż profesjonalnego sprzętu komputerowego i peryferii.

1. JĘZYKI PROGRAMOWANIA, BIBLIOTEKI, GRAFIKA

Basic programowanie w języku Basic cz. III	52 000 zł
Turbo Basic	183 000 zł
Proffesional Fortran + dodatki	144 000 zł
Fortran 77	132 600 zł
Fortran 80	65 300 zł
Język C zastosowanie dla zaawansowanych	214 200 zł
Kompilator języka C	106 000 zł
System Turbo C V.1.5	79 000 zł
Turbo C biblioteka	150 000 zł
Turbo C podręcznik użytkownika	109 000 zł
Turbo Pascal	123 500 zł
Turbo Pascal V.4.0	277 000 zł
Turbo Pascal V.5.0	161 000 zł
Turbo Pascal V.5.5 dodatek do T.P.V.5.0	56 000 zł
Turbo Power Tools Plus	124 000 zł
Turbo Database Toolbox	33 000 zł
Modula 2 86	125 500 zł
Turbo Pascal 5.0 opis języka	189 000 zł
Turbo Pascal 4.0 (aut. Jan Bielecki)	153 000 zł

2. BAZY DANYCH, PAKIETY ZINTEGROWANE

Clipper do dBase III	30 000 zł
Clipper do dBase III +	131 000 zł
Clipper 86	158 000 zł
Clipper 87	212 700 zł
Clipper 87 do dBase III +	214 000 zł
dBase II i III podręcznik dla zaawansowanych	205 000 zł
dBase III + instalacja pakietu relacyjnej bazy danych	22 000 zł
dBase III + zastosowanie	205 000 zł
dBase III + opis pakietu sieciowego	55 000 zł
dBase III + programowanie	126 000 zł
dBase III + poznawanie	111 000 zł
dBase III poradnik encyklopedyczny	237 000 zł
dBase IV	249 000 zł
Framework II P	162 000 zł
Informix	261 000 zł
Lotus 1 2 3 + dodatek (komplet)	113 000 zł
Multiplan	66 000 zł
Pakiet C-ISAM	34 700 zł
CX Fortran kompilator języka do systemu DOS2	114 300 zł
Plib 86/Plink 86 opis języka	47 000 zł
Statgraphics	91 000 zł
Programowanie systemów baz danych języka Clipper	107 000 zł
Kompilator CBasic tom. II	92 000 zł

3. SYSTEMY OPERACYJNE, PROGRAMY UŻYTKOWE

System operacyjny DOS cz. 1 i 2	155 000 zł
System operacyjny DOS 3.10	47 000 zł
System operacyjny DOS 3.20	181 000 zł
System operacyjny DOS 3.30	180 000 zł
System operacyjny DOS 4.0	202 000 zł
Eureka	64 000 zł
Norton Commander	63 000 zł
Polwindows	50 000 zł
Sidekick	44 000 zł

4. EDYTORY

Chl-2 Writer	31 000 zł
Personalny Edytor PE	68 300 zł
Wordstar 2000	142 000 zł

5. CAD-SIECI

AutoCad 2.17	137 000 zł
AutoCad	30 600 zł
Lanlink 4.0 sieć lokalna	80 000 zł
NOWELL, instalacja sieci (Arcnet)	9 000 zł
NOWELL, instalacja sieci	65 000 zł
NOWELL, podręcznik użytkowania sieci	116 000 zł
Or-Cad	134 000 zł

6. ROZMAITOŚCI

Przewodnik programisty	205 000 zł
Printer STAR NL-10 instrukcja	20 000 zł
Wprowadzenie do użytkowania komputerów osobistych klasy IBM PC	43 000 zł
Drukarka NX-16 instrukcja	106 000 zł
Drukarz	14 000 zł
Karta system rozliczania kart drogowych	10 000 zł
Appsgen	18 000 zł

Zamówienia prosimy kierować pod adresem:
INTERSOFT Sp. z o.o.
00-496 Warszawa, ul. Krucza 16/22,
telefony: 28-44-81 w. 284, 352; 28-29-53,
teleks: 812414

0/7/90

UWAGA UŻYTKOWNICY SYSTEMÓW KOMPUTEROWYCH PRACUJĄCYCH POD KONTROLĄ S.O. UNIX!

PRZEDSIĘBIORSTWO TECHNICZNO-HANDLOWE
„KaNet" Sp. z o.o.
41–500 CHORZÓW, ul. Katowicka 77
tel. 412-122, teleks 312627 kanet pl

oferuje:

■ Oprogramowanie narzędziowe pakietu autorskiego mixPACK:

☆ **mixCOMM** – **program wspomagający zarządzanie plikami:**
rozwinięcie koncepcji programu NORTON COMMANDER, umożliwiające pracę w środowisku wielodostępnym lub sieci lokalnej (s.o. UNIX, XENIX i pochodne);

☆ **mixEDIT** – **pełnoekranowy edytor tekstowy:**
wzorowany na programie NORTON EDITOR, przeznaczony m.in. do pracy w środowisku wielodostępnym lub sieci lokalnej (s.o. UNIX, XENIX i pochodne, MS DOS), ukierunkowany na prace programistyczne;

☆ **mixOFFI** – **pakiet wspomagający prace biurowe:**
implementacja koncepcji pakietów zintegrowanych dla systemów operacyjnych UNIX System V lub pochodnych (w tym XENIX 286 i 386 firmy SCO) oraz systemu DOS;
uniwersalny kalendarz wraz z terminarzem, arkusz kalkulacyjny, baza danych o parametrach definiowanych przez użytkownika, tabela kodów ASCII.

■ Oprogramowanie aplikacyjne z zakresu organizacji i zarządzania przedsiębiorstwem,

■ Nowoczesne rozwiązania sprzętowe firmy Texas Instruments w dowolnych konfiguracjach.

POSIADAMY PRAWA AUTORSKIE NA CAŁOŚĆ OFEROWANEGO OPROGRAMOWANIA!

GWARANTUJEMY WIELOLETNIE UŻYTKOWANIE ZAKUPIONEGO OPROGRAMOWANIA DZIĘKI MOŻLIWOŚCI PRZENOSZENIA GO NA DOWOLNE SYSTEMY KOMPUTEROWE!

ZAPRASZAMY DO KORZYSTANIA Z NASZEJ OFERTY!

0/16/90

EWA GRABSKA
GRAŻYNA HLINIAK
JACEK LEMBAS
Katedra Informatyki UJ
Kraków

Zastosowanie grafów kompozycyjnych do konstrukcji strukturalnego edytora graficznego

W ostatnim czasie coraz powszechniejsze staje się wykorzystanie grafiki w programowaniu, czyli tzw. programowanie wizualne. Można przytoczyć kilka argumentów przemawiających za używaniem grafiki zamiast stosowanego dotychczas tekstu. Informacja zawarta w tekście ma ścisły, liniowy porządek, natomiast informacja zawarta w rysunku jest dostępna w sposób swobodny. W grafice istnieje więcej środków do uwidocznienia istotnych informacji (kształt, rozmiar, kolor, rozmieszczenie przestrzenne). A co najważniejsze, pamięć ludzka ma naturę graficzną o czym świadczyć może fakt, że większość ludzi łatwiej zapamiętuje to, co widzi, niż to, co słyszy.

Nasz mózg umożliwia myślenie wizualne i dlatego użycie grafiki stymuluje nasze zdolności twórcze. Stare przysłowie mówi, że *rysunek wart jest tysiąca słów*. Dodatkowo dla obrazów nie istnieje bariera językowa. Dobrze zaprojektowane piktogramy mogą być prawidłowo odczytywane i rozumiane przez ludzi różnych narodowości.

Z drugiej strony, źle zinterpretowany obraz (podobnie jak tekst) może łatwo prowadzić do nieporozumień. Nie jest to jednak wina grafiki, lecz niezdefiniowania semantyki używanych symboli lub niejednoznaczności ich przestrzennego rozmieszczania. Wyposażenie grafiki w dokładny syntaktyczny i semantyczny opis pozwala uniknąć tego typu błędów. Zdefiniowanie takiego opisu dla klasy grafiki nazywa się językiem diagramowym, a każdą grafikę, dla której istnieje syntaktyczny i semantyczny opis – *diagramem* [2].

W wielu zastosowaniach informatyki używa się systemów opartych na diagramach np. SADT *(Structured Analysis and Design Technique)* do analizy [5], JDM *(Jackson Design Methodology)* lub HIPO *(Hierarchy plus Input-Process-Output)* do projektowania systemów informatycznych [2], oraz schematów Nassi-Shneidermana do kodowania [6].

Diagramy są wykorzystywane również jako wizualna ilustracja algorytmów, odzwierciedlając ich strukturę. Obecnie próbuje się konstruować diagramy, które mogłyby być bezpośrednio wykonywane przez komputer. Programy takie są łatwiejsze do zrozumienia, modyfikowania i dokumentowania.

Ważnym problemem związanym z językiem diagramów jest opracowanie interakcyjnego interfejsu graficznego, zwanego edytorem graficznym, który umożliwia rysowanie i modyfikowanie diagramów. Najczęściej konstruuje się edytory sterowane syntaktycznie, pozwalające użytkownikowi rysować tylko diagramy poprawne syntaktycznie.

W naszym artykule przedstawimy koncepcję systemu projektowania wspomaganego komputerowo LEONARDO, w którym wykorzystany zostanie graficzny edytor diagramów BFE. Edytor ten może pracować w dwóch trybach. W jednym z nich działa jako edytor syntaktyczny sterowany składnią, sprawdzając poprawność syntaktyczną diagramów w każdym kroku ich konstrukcji. Drugi tryb pozwala użytkownikowi tworzyć dowolne diagramy, których poprawność jest weryfikowana po ich narysowaniu.

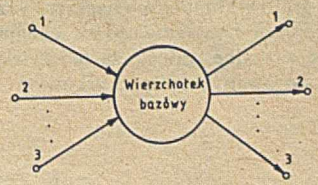
Edytory graficzne wymagają zdefiniowania: składowych diagramów, dostępnych użytkownikowi funkcji stosowanych podczas konstrukcji oraz syntaktycznych reguł łączenia składowych.

Diagramy mogą być reprezentowane za pomocą atrybutowych grafów. Atrybuty określają między innymi współrzędne położenia poszczególnych składowych diagramu. Atrybutowane grafy zawierają informację potrzebną do odtworzenia diagramów i stanowią ich wewnętrzną reprezentację w komputerze. Grafowym odpowiednikiem języków diagramów są języki grafów atrybutowanych, do definiowania których używa się gramatyk grafowych. Gramatyki te będą tematem następnego rozdziału.

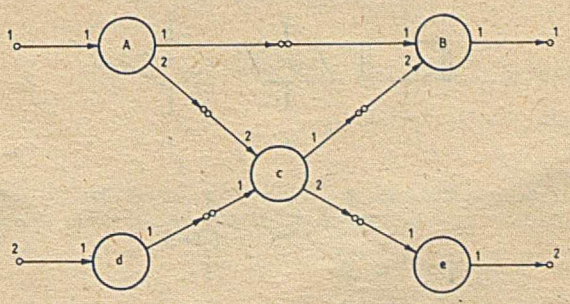
Gramatyki grafów kompozycyjnych

Istnieje wiele różnych definicji gramatyk grafowych [1]. Ta różnorodność jest motywowana różnorodnością zastosowań tej koncepcji. Nie istnieje jednak przynajmniej obecnie tendencja do wybrania jednego typu gramatyk grafowych, w terminach której wszystko powinno być zdefiniowane. W tym artykule proponujemy nową gramatykę grafową, która naszym zdaniem może znaleźć zastosowanie w definiowaniu języków diagramów. Zdefiniowana gramatyka nazywa się gramatyką grafów kompozycyjnych.

Grafy kompozycyjne definiuje się na podstawie hierarchicznej struktury grafowej. Rozważa się skierowane, uporządkowane grafy etykietowane, w których wierzchołki mogą być także grafami. Definiuje się pojęcie grafu elementarnego (E-grafu) składającego się z dwóch wierzchołków: wierzchołka bazowego oraz wierzchołka otoczenia, który jest grafem. Krawędzie łączące wierzchołek bazowy z wierzchołkiem otoczenia nazywa sią wiązaniami. Istnieją dwa rodzaje wiązań: wiązania wejściowe i wiązania wyjściowe. Graficzna reprezentacja E-grafu (rys. 1) zawiera tylko wierzchołek bazowy wraz z wiązaniami.



Rys. 1



Rys. 2

Grafy kompozycyjne (C-grafy) są budowane przez łączenie grafów elementarnych. Połączenia są ustalane między wiązaniami wyjściowymi i wiązaniami wejściowymi. Rys. 2 przedstawia przykład C-grafu.

Dla grafów kompozycyjnych te wiązania, które są połączone, nazywają się wewnętrznymi. Pozostałe wiązania są zewnętrzne. W ten sposób otrzymuje się naturalną hierarchiczną strukturę grafów kompozycyjnych, w której graf kompozycyjny może być rozważany jako graf elementarny na wyższym poziomie formalizmu [3].

Języki grafów kompozycyjnych mogą być zdefiniowane za pomocą gramatyk grafów kompozycyjnych. Gramatyka C-grafów jest systemem zawierającym następujących sześć elementów:
1) zbiory *etykiet końcowych* i *etykiet pomocniczych wierzchołków bazowych*,
2) zbiory *etykiet końcowych* i *etykiet pomocniczych wiązań*,
3) zbiór *E-grafów nieterminalnych*, to znaczy takich, w których wierzchołki bazowe mają etykiety pomocnicze, a wiązania są nieetykietowane,
4) zbiór *E-grafów terminalnych*, to jest takich, w których wierzchołki bazowe oraz wiązania mają etykiety końcowe,
5) skończony zbiór *produkcji*, które są parami następującej postaci: *(lewa strona, prawa strona)*, gdzie lewa strona jest E-grafem nieterminalnym, a prawa strona jest C-grafem o tej samej liczbie wiązań wejściowych i wyjściowych co strona lewa, oraz zbudowanym z E-grafów terminalnych lub/i E-grafów nieterminalnych,
6) *symbol startowy będący* C-grafem lub E-grafem i składający się z E-grafów nieterminalnych.

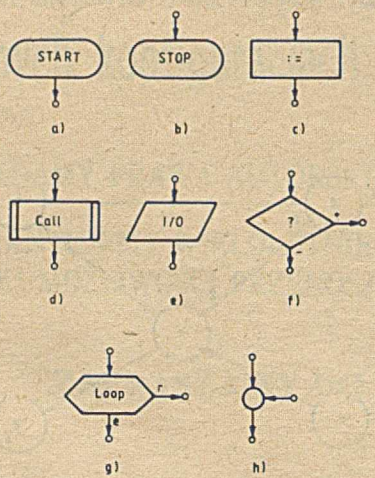
W następnym rozdziale omówiona będzie gramatyka C-grafów do konstrukcji strukturalnych schematów blokowych. Na rys. 3 przedstawione są E-grafy terminalne tej gramatyki. Natomiast rys. 4 pokazuje jej E-grafy nieterminalne. Zbiór produkcji tej gramatyki składa się z jedenastu produkcji (rys. 5). E-grafy, będące lewymi stronami produkcji, są reprezentowane za pomocą prostokątów, z których wychodzą i do których wchodzą ich wiązania. Etykiety nieterminalne wierzchołków bazowych tych E-grafów znajdują się w lewym górnym rogu. Wewnątrz prostokątów umieszczone są C-grafy będące prawymi stronami produkcji.

Generowanie C-grafów opiera się na operacji zastępowania w danym C-grafie jego E-grafów przez odpowiadające im C-grafy. Zastępowane E-grafy są izomorficzne (mają jednakowe struktury i etykiety) z lewymi stronami takich produkcji, których prawe strony są C-grafami zastępującymi te E-grafy.

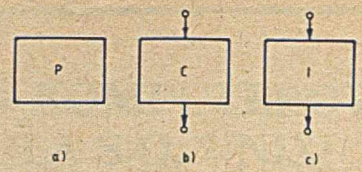
Językiem generowanym przez gramatykę nazywamy zbiór wszystkich C-grafów, które można wygenerować z symbolu startu i które są zbudowane z E-grafów terminalnych.

ANALIZA GRAFÓW KOMPOZYCYJNYCH

Problemy analizy grafów kompozycyjnych rozważymy na przykładzie schematów blokowych. Diagramy będące schematami blokowymi, ze względu na swoje wizualne walory, są najczytelniejszym zapisem algorytmów często stosowanym w początkowym nauczaniu informatyki.



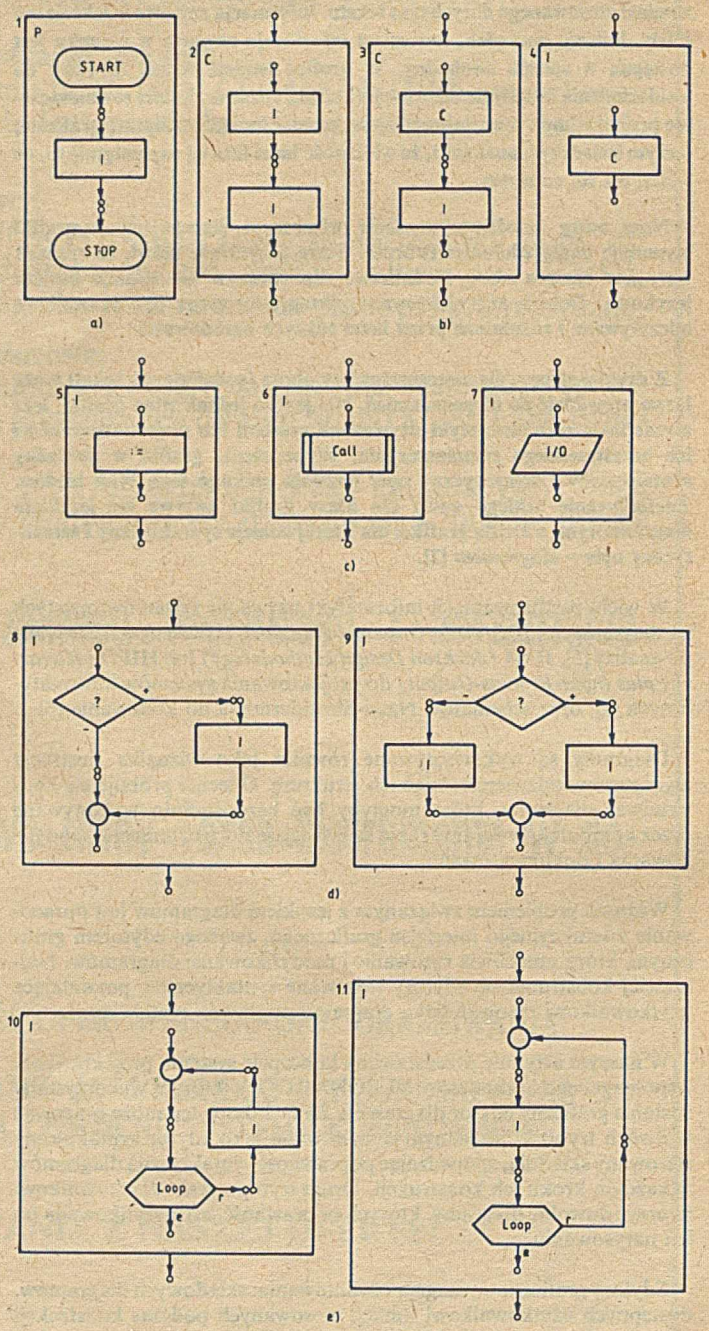
Rys. 3



Rys. 4

Prezentowana przez nas teoria grafów kompozycyjnych doskonale nadaje się do formalnego zapisu schematów blokowych. Zbiór terminalnych E-grafów przedstawiony jest na rys. 3. Na rysunku tym przedstawione są początek i koniec schematu (rys. 3a i 3b), instrukcja przypisana (rys. 3c), wywołanie procedury (rys. 3d), operacje wejścia–wyjścia (rys. 3e), instrukcja warunkowa z wyjściami tak i nie (rys. 3f), instrukcja iteracyjna z wyjściami powtarzania i zakończenia (rys. 3g) oraz łącznik dwóch dróg algorytmu (rys. 3h).

Z doświadczenia wiadomo, że budowa schematów blokowych pozostawia konstruktorowi dużo swobody nie narzucając żadnych ograniczeń na sposób połączeń w schemacie. Jednym z wymagań w początkowym nauczaniu informatyki jest strukturalność konstruowanych sche-



Rys. 5

matów, która daje możliwość ich hierarchicznej budowy według ściśle określonych reguł. Do przedstawienia tych reguł bardzo dobrze nadają się gramatyki grafów kompozycyjnych.

Elementy schematów opisane za pomocą nieterminalnych E-grafów są przedstawione na rys. 4 i oznaczają kolejno: program (symbol startowy – rys. 4a), ciąg instrukcji (rys. 4b) oraz pojedynczą instrukcję (rys. 4c).

Natomiast gramatyka grafów kompozycyjnych opisująca reguły strukturalizacji może zawierać zbiór produkcji przedstawiony na rys. 5.

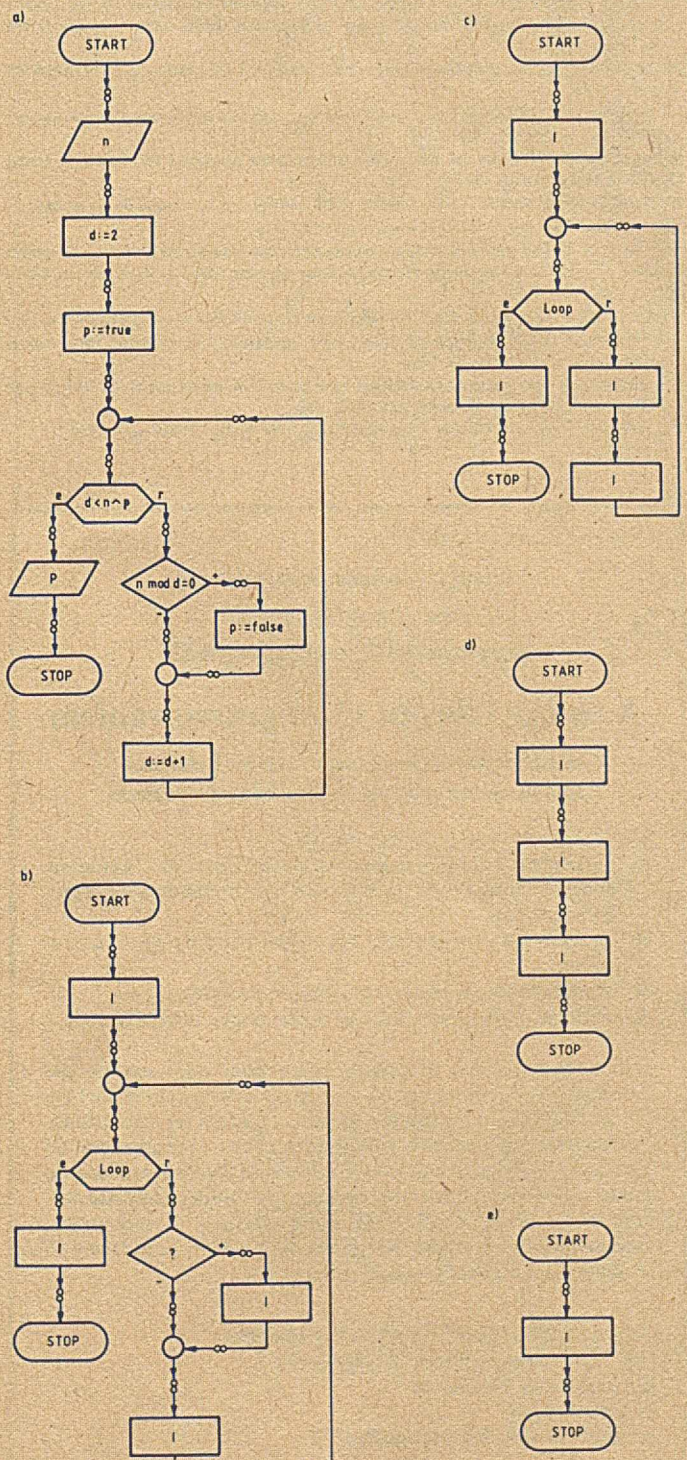
Gramatyka o przedstawionym zbiorze produkcji może być użyta do konstrukcji strukturalnych schematów blokowych. Jej istotnym zastosowaniem będzie sprawdzenie, czy skonstruowany schemat blokowy jest strukturalny. W tym wypadku gramatyka używana jest do redukcji istniejącego schematu. Istnieją dwie możliwości postępowania: *one line* – dopuszczająca tylko strukturalne połączenia w czasie tworzenia schematu oraz *off line* – sprawdzająca strukturalność po skonstruowaniu schematu. Proces redukcji przedstawimy na następującym przykładzie. Skonstruujmy schemat blokowy algorytmu sprawdzającego, czy dana liczba naturalna $n > 1$ jest pierwsza.

Dana : n ∈ ℕ, n > 1,

Zmienna pomocnicza : d ∈ ℕ – aktualny dzielnik,

Wynik :

$$p = \begin{cases} true & \text{gdy n liczba pierwsza} \\ false & \text{gdy n liczba złożona} \end{cases}$$

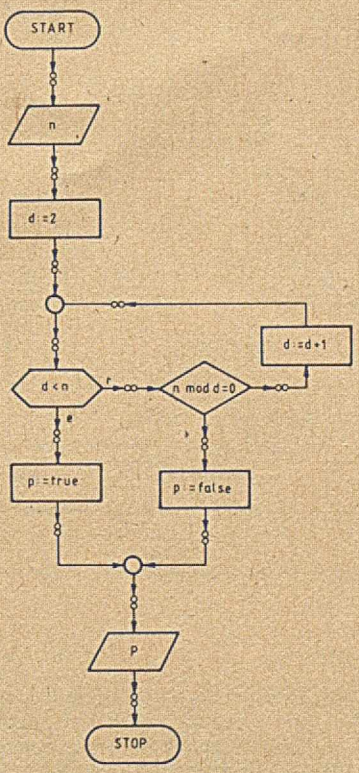


Rys. 6

Przebieg poprawnego procesu redukcji przedstawiony jest na rys. 6.

Schemat blokowy algorytmu oraz kolejne schematy, będące wynikami redukcji, reprezentowane są za pomocą grafów kompozycyjnych. Stosując do schematu blokowego z rys. 6a redukcje produkcjami 2–7 z rys. 5b i 5c otrzymujemy schemat pokazany na rys. 6b. Redukcja produkcją 8 (rys. 5d) daje schemat z rys. 6c. Teraz stosujemy redukcje produkcjami 2 i 10 (rys. 5b i 5e), w wyniku których otrzymujemy schemat z rys. 6d. Ostateczny wynik redukcji, po zastosowaniu kolejno produkcji 2, 3 i 4 (rys. 5b), przedstawiony jest na rys. 6e.

Częstym błędem początkujących informatyków, zaburzającym strukturę schematu blokowego, jest zastosowanie niespodziewanego „wyskoku" z pętli. Przedstawiony przykład stwarza taką możliwość, a niestrukturalny schemat może wyglądać jak na rys. 7.



Rys. 7

Po początkowych redukcjach, zastępujących instrukcje elementarne E-grafem nieterminalnym z rys. 4c, nie można znaleźć produkcji redukującej jakikolwiek fragment schematu zawierający jedną z instrukcji warunkowych. Reakcją analizatora składni programu dokonującego automatycznej redukcji – na taki rodzaj błędu będzie komunikat, że nie istnieje produkcja pozwalające zredukować wykryty fragment schematu.

Implementacja edytora graficznego BFE

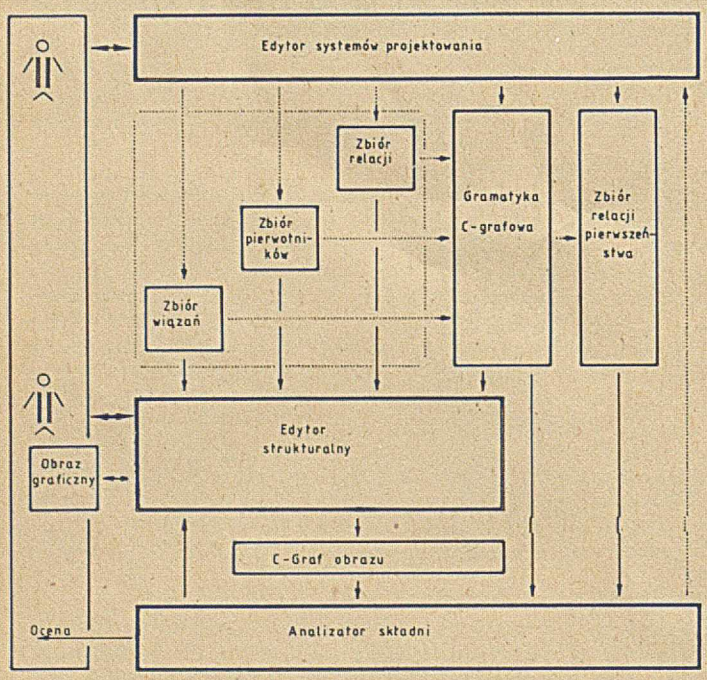
Celem skonstruowania edytora graficznego BFE było zbadanie możliwości zastosowania teorii grafów kompozycyjnych do opisu struktur. Edytor został zaimplementowany w języku Turbo Pascal na mikrokomputerze typu IBM PC. Jednym z powodów wyboru tego typu sprzętu była powszechna jego dostępność w Polsce, a co za tym idzie, duża liczebność ewentualnych, przyszłych użytkowników BFE.

BFE jest przeznaczony do składania schematów blokowych i przygotowywania pewnego rodzaju ilustracji, które mogą być wkomponowane w tekst publikacji naukowych napisanych w TEXu. Nie są to jednak jedyne dziedziny, w których można zastosować tego typu strukturalny edytor graficzny.

Przewidujemy, że BFE zostanie użyty między innymi do opisu schematów elektrycznych i elektronicznych oraz innych instalacji przemysłowych.

LEONARDO – Komputerowy system wspomagający projektowanie

Edytor graficzny BFE stanowi jeden z modułów komputerowego systemu wspomagającego projektowanie LEONARDO, przedstawionego na rys. 8. W systemie tym można tworzyć diagramy (potencjalne rozwiązania projektowe) w sposób interakcyjny oraz sprawdzać czy spełniają one przyjęte kryteria projektowe. W systemie LEONARDO przetwarzanie informacji polega na manipulowaniu strukturami wyższych wymiarów, opisanymi za pomocą grafów kompozycyjnych.



Rys. 8

Oprócz edytora BFE system ten zawiera dwa dodatkowe moduły: edytor projektowania systemów oraz moduł analizatora. Pierwszy z nich umożliwia projektowanie kształtów pierwotników graficznych i tworzenie ich biblioteki oraz definiowanie gramatyki grafów kompozycyjnych wraz z wyznaczaniem relacji pierwszeństwa. Natomiast moduł analizatora pozwala na redukcję C-grafów reprezentujących diagramy utworzone za pomocą edytora graficznego BFE. Algorytm analizatora umożliwia odpowiedź na pytanie, czy redukcja jest możliwa, a w przypadku pozytywnym, wybór redukowanej części i jej zastąpienie częścią bardziej ogólną, tak jak to było opisane w rozdziale dotyczącym analizy grafów kompozycyjnych.

LITERATURA

[1] Ehrig H., Nagl M., Rosenberg G., Rosenfeld A. (eds): Graph-Grammars and their Application to Computer Science. Lect. Not. Comp. Sci. 291, 1987

[2] Goettler H., Grabska E.: Attributed Graph-Grammar and Composition Nets. Proc. (WG' 85) Int. Workshopp on „Graphtheoretic Concepts in Computer Science", Trauner Linz, 1985

[3] Grabska E.: Some Properties of CP-net Concatenation. Fourth International Workshop on Graph Grammars and their Application to Computer Science, Proceedings, Bremen 1990

[4] Jackson M. A.: Principles of Program Design. Academic Press, London, 1975

[5] Ross D. T.: Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transactions on Software Engineering, Vol. S. SE-3, No. 1 (Jan.) 1977

[6] Shu N. C.: Visual Programming. Van Nostrand Reinhold Company, New York, 1988.

Problemy współczesnej rekonstrukcji obrazów

dokończenie ze s. 21

[5] Norton A., Silberger A.J.: Parallelization and performance analysis of the Cooley-Tukey FFT algorithm for shared-memory architectures. IEEE Trans. Comput., Vol. C-36, No. 5, 1987

[6] Nowiński W.L.: Rekonstruowanie obrazów w tomografii komputerowej. INFORMATYKA nr 3, 1989

[7] Nowiński W.L.: Algorytmy rekonstruowania obrazów za pomocą metod transformacyjnych. INFORMATYKA, nr 4, 1989

[8] Nowiński W.L.: Algorytmy rekonstruowania obrazów metodami rozwinięcia w szereg. INFORMATYKA, nr 5, 1989

[9] Nowiński W.L.: Parallel implementation of the convolution method in image reconstruction. Lecture Notes in Computer Science, September 1990

[10] Nowiński W.L.: Współbieżna implementacja algorytmu rekonstrukcji obrazów. Materiały Konferencji „Grafika Komputerowa i Przetwarzanie Obrazów" GKKO'90, Kazimierz Dolny 1990

[11] Nowiński W.L.: Asynchronism in parallel image reconstruction. Prace IPI PAN, 676, 1990

[12] Peschmann K.R. i in.: High-speed computed tomography: systems and performance. Applied Optics, Vol. 24, No. 23, 1985

[13] Ritman E.L. i in.: Physics and technical considerations in the design of the DSR: a high temporal resolution volume scanner. AJR, 134, 1980

[14] Tataj M., Toruń A.: Tomografia komputerowa. INFORMATYKA nr 7, 1989.

Uwaga Twórcy, Konsultanci oraz zainteresowani programami komputerowymi!!!

Agencja Obrotu Oprogramowaniem

działająca przy Wydawnictwach Naukowo-Technicznych

to nowość na polskim rynku informatycznym

Główne formy naszej działalności to:

- wydawanie katalogu zawierającego informacje o aktualnie dostępnych na rynku programach oraz ogłoszenia na zamówienie klienta;
- rejestrowanie programów dla celów ochrony praw autorskich;
- pośredniczenie w sprzedaży oprogramowania;
- organizowanie przetargów na rozwiązania informatyczne.

Jeśli chcesz wybrać programy najbardziej odpowiednie dla Twojego przedsiębiorstwa, nie musisz już dzwonić do wielu firm ani jeździć po całym kraju w celu zapoznawania się z programami. My mamy je wszystkie w jednym miejscu a nasz adres jest do Twojej dyspozycji. Pomożemy Ci dokonać wyboru w sposób o jakim marzyłeś od dawna; nie ruszając się z biura naszej Agencji. Jeśli masz programy – efekt wielu miesięcy Twojej pracy, zarejestruj je u nas zanim piraci je skopiują, a my pomożemy Ci w ochronie Twoich praw autorskich.

Agencja Obrotu Oprogramowaniem
przy Wydawnictwach Naukowo-Technicznych
ul. Mazowiecka 2/4 pok. 206
00-048 Warszawa
tel. 26-71-71 wew. 304, telefaks 26-82-93

0/3/90

MEDCOM

ZASILANIE AWARYJNE

ZASILACZE BEZPRZERWOWE

naszej produkcji

UPS – 450

produkcji tajwańskiej

UPS–500

produkcji japońskiej

GENERATORY PRĄDOTWÓRCZE

różnych mocy

EO 3 90

Biuro
Al. Ujazdowskie 26/39
00–478 Warszawa tel. 28 93 57, teleks 81 70 60

ZAKŁAD PRODUKCYJNO-USŁUGOWY
ul. Wołodyjowskiego 41
02–724 Warszawa tel.43 04 41

Jak startuje UNIX

Zacznijmy od wyznaczenia punktów granicznych: chodzi o przedział czasu między włączeniem maszyny a rozpoczęciem sesji użytkowej. Zrozumienie tego, co się wtedy dzieje, pozwala w znacznym stopniu wpływać na poszczególne etapy tego procesu. Możemy wręcz doprowadzić do tego, że system będzie się zachowywał na pozór zupełnie nie jak UNIX. Jest to przydatne, gdy tworzymy system użytkowy „pod klucz" i nie interesują nas standardowe metody obsługi użytkowników. Niektóre, zwłaszcza wstępne etapy, nie dają się zmodyfikować i o nich wspomnę tylko w skrócie. Ilustracje czerpać będę głównie z najbardziej popularnej na świecie wersji UNIX-a, czyli systemu SCO XENIX System V.

Najpierw powoli, jak żółw ociężale

Jądro systemu rezyduje na dysku w zwykłym pliku o nazwie związanej z nazwą systemu (unix, xenix, vmunix itp.). Plik ten umieszczony jest najczęściej w nadrzędnym skorowidzu głównego systemu plików. Trzeba ten plik znaleźć, załadować do pamięci i przekazać mu sterowanie. Zajmuje się tym maszynowo zależny program o nazwie **boot**. Program **boot** miewa zwykle parametry, za pomocą których można np. zamienić plik zawierający jądro na inny lub zadać inny system plików jako główny. Parametry te możemy wpisać z konsoli na żądanie programu **boot** po włączeniu maszyny lub też mogą one być zapisane w innym pliku (w XENIX-ie jest to /etc/default/boot). Wynika stąd jedna ważna praktyczna cecha: możliwe jest uruchamianie różnych wariantów systemu operacyjnego na tej samej instalacji sprzętowej, na przykład zawierających różne programy obsługi tego samego urządzenia lub inaczej skonfigurowane bufory czy inne zasoby jądra systemu.

Jądro UNIX-a ma skłonność do tycia: kolejne, nowe wersje systemu z różnych jego linii rozwojowych są coraz bardziej opasłe – realizują coraz więcej funkcji, kod staje się coraz bardziej złożony a z dawnych funkcji najczęściej nic nie ubywa. Plik przeznaczony do załadowania do pamięci jest więc duży, a algorytm operacji dyskowych w programie **boot** uproszczony. Sprawia to, że na załadowanie musimy trochę poczekać.

Po otrzymaniu sterowania jądro systemu musi wykonać kilka podstawowych czynności. Można wśród nich wymienić ustawienie odpowiedniego kontekstu głównego procesora (rejestry stanu, stosu itp.) oraz rozpoznanie pamięci operacyjnej i przygotowanie struktury danych do obsługi pamięci wirtualnej.

Szarpnęła wagony i ciągnie z mozołem

Po wykonaniu wstępnych czynności jądro zaczyna inicjalizować **programy obsługi urządzeń** (ang. *device drivers*). Niemal każdy taki program, wśród funkcji, które realizuje (takich jak open( ), close( ), read( ), write( )) zawiera również funkcję init( ). Jest ona wywoływana dokładnie raz właśnie na tym etapie rozruchu systemu. Autor programu obsługi ma więc możliwość zadania w tej funkcji takich czynności, jak np. załadowania rejestrów sterowników czy dynamicznej inicjalizacji struktur. Nie należy domniemywać kolejności, w jakiej są wołane funkcje init( ) kolejnych programów obsługi urządzeń.

Najczęściej w tym momencie, a czasem w następnym kroku rozruchu systemu sprawdzany jest główny system plików. Jeśli UNIX nie był poprzednio zamknięty w sposób właściwy, system plików może wymagać naprawy, do czego służy program **fsck**.

Gdy już mamy zainicjalizowane programy obsługi urządzeń możemy z grubsza stwierdzić, że faza ładowania i uruchamiania jądra zakończyła się. Następnym etapem jest inicjalizacja systemu procesów.

UNIX jest zastrzeżonym znakiem towarowym AT&T.

I biegu przyspiesza, i gna coraz prędzej

Istnieje zwykle kilka procesów uprzywilejowanych. Niektóre z nich to procesy systemowe, których kod zawarty jest w ciele jądra systemu. Jednym z takich procesów jest **swapper** – proces odpowiedzialny za wymianę obrazów pamięci nieaktywnych procesów między pamięcią operacyjną i jej przedłużeniem na dysku. Jednak najważniejszym i najbardziej interesującym procesem uprzywilejowanym jest **init**. Ma on zawsze numer 1, istnieje przez cały czas działania systemu i jest przodkiem wszystkich innych procesów „zwykłych". Proces ten jest zapisany w pliku /etc/init. Uprzywilejowanie jego polega na tym, że nie można go zabić. Poza tym jest traktowany jak każdy inny proces, którego rzeczywistym i efektywnym użytkownikiem jest **root**. Proces ten może więc realizować niemal dowolny program, który sobie napiszemy i umieścimy w /etc/init. Pamiętać trzeba tylko o jednym: proces nr 1 nie ma standardowego otoczenia, do jakiego jesteśmy przyzwyczajeni. Nie możemy np. odwoływać się do klasycznych pojęć wejścia/wyjścia **stdin** i **stdout**, gdyż init nie ma żadnych otwartych plików. Musimy dopiero sami je otworzyć.

Od tego momentu kończy się więc styl pracy, który UNIX wymusza a zaczyna się sposób działania, który jest proponowany użytkownikowi jako standardowy.

A dokąd? A dokąd? A dokąd? Na wprost!

Standardowy program **init** w mniej lub bardziej otwarty sposób daje użytkownikowi do wyboru dwie drogi: zwykłą pracę wieloużytkową na dołączonych terminalach lub jednostanowiskową pracę na konsoli dla administratora systemu. Ten drugi tryb jest konieczny, gdy mamy do wykonania prace typu porządkowego, przy których niewskazana lub niemożliwa jest jednoczesna praca wielu użytkowników. Do realizacji tego trybu init powołuje proces realizujący program /bin/sh, czyli najbardziej popularny w UNIX-ie interpretator poleceń. Po zakończeniu tego trybu system przechodzi do pracy wielostanowiskowej.

Wywoływany jest znowu /bin/sh, ale tym razem nie w trybie interakcyjnym, lecz dla wykonania programu /etc/rc w języku **Shell**. Czasem jest to kilka podprogramów; są wówczas umieszczone bądź w skorowidzu /etc/rc.d, bądź w skorowidzu o nazwie zbliżonej.

Przez góry, przez tunel, przez pola, przez las

Program (programy) /etc/rc ma do wykonania kilka typowych czynności. Przez to, że jest to plik tekstowy, możemy łatwo ingerować w jego treść, usuwać lub dodawać nowe czynności. Jest to najprostszy – i zalecany sposób – w jaki administrator może wpływać na rozruch systemu. Typowe, zwykle wykonywane przez /etc/rc czynności, to:

- ustawienie daty i czasu rzeczywistego (zegara),
- zamontowanie dodatkowych systemów plików,
- uruchomienie procesów działających w tle, tzw. demonów, z których najważniejsze to **cron** i „spooler" drukarek,
- wyczyszczenie skorowidzów roboczych.

Po zakończeniu pracy /etc/rc sterowanie wraca ponownie do procesu init.

Gładko tak, lekko tak toczy się w dal

Teraz init powołuje tyle procesów, ile jest zadeklarowanych aktywnych linii komunikacyjnych (terminali) w danej instalacji. Linie te są opisane w tekstowych bazach danych, z których wszędzie spotykanym jest plik /etc/ttys, natomiast XENIX używa jeszcze do tego plików /etc/ttytype i /etc/gettydefs.

dokończenie na s. 32

Systemy otwarte i ich wrogowie, czyli co dalej z UNIX-em (2)

Główni konkurenci

W rozgrywce dwóch UNIX-owych rywali chwilowo walkowerem prowadzi **AT&T UNIX System V Release 4 (SVR4)**. O systemie operacyjnym **OSF/1** – który spóźniając się traci punkty – wiadomo wciąż niewiele ponad to, że prawdopodobnie będzie łączył budowę wewnętrzną eksperymentalnego systemu **Mach** z cechami użytkowymi IBM-owskiego **AIX**-a. Dlatego z konieczności przedstawiając bliżej tylko SVR4, nieco uwagi poświęcimy również tym dwóm systemom.

UNIX System V Release 4

Kolejne edycje UNIX-a z AT&T zawsze wzbudzają zainteresowanie. Ale SVR4 rzeczywiście na nie zasługuje: chyba nigdy dotąd nie podjęto tak daleko idącej próby unifikacji UNIX-a i nie wprowadzono na raz tylu nowości. A oto szczegóły:

Modularne jądro. Jak wiadomo, UNIX nie ma rozbudowanej struktury warstwowej – w oprogramowaniu UNIX-owym wyraźnie widać tylko dwie warstwy: system (jądro) i resztę. Do tej reszty należą programy użytkowe, a więc również programy usługowe i powłoka (ang. *shell*), która ma status zwykłego programu użytkowego. Natomiast sam system operacyjny niemal w całości zawiera się w jądrze (ang. *kernel*), które właściwie nie ma warstwowej budowy. Wraz z rozwojem UNIX-a jądro powiększało się coraz bardziej – na przykład o mechanizmy sieciowe, często ze szkodą dla prostoty i spójności (w SVR4 jest o 1/4 większe niż w SVR3.2 a system wymaga 4 MB pamięci). Na szczęście jądro SVR4 wreszcie stało się modularne, czego efektem jest lepsza konfigurowalność i przenośność systemu.

Wirtualny system plików. Klasyczny UNIX-owy system plików – wielokrotnie krytykowany – od dawna podlegał rozmaitym modyfikacjom, które zwiększały jego niezawodność, efektywność czy funkcjonalność. Wprowadzano dodatkowe egzemplarze superbloku, zmieniano wielkość bloku, dzielono bloki na mniejsze fragmenty, listy wolnych bloków zastępowano mapami bitowymi, dodając węzły wirtualne uogólniano mechanizm *i*-węzłów, itd. Potem przyszła kolej na rozszerzenia sieciowe – protokoły umożliwiające dostęp do plików nielokalnych i ich współdzielenie – oraz mechanizmy dołączania obcych systemów plików. *Wirtualny system plików* SVR4 pozwala korzystać z niemal wszystkich wymyślonych dotąd udoskonaleń. Powstał w wyniku rozszerzenia systemu plików z SVR3.2 o rozwiązania pochodzące z SunOS (np. *v*-węzły) i jest zgodny z tak zwanym *szybkim systemem plików* z 4.3BSD. Umożliwia korzystanie z rozszerzeń sieciowych – *Network File System* (NFS) z SunOS i *Remote File Sharing* (RFS) z AT&T – oraz z własnych systemów plików użytkownika. Wprowadzono wyspecjalizowane systemy plików użytkownika. Wprowadzono wyspecjalizowane systemy plików dla plików FIFO (*FIFOFS*) i plików urządzeń (*SPECFS*). Rozbudowano środki ryglowania plików i rekordów. Z 4.3BSD przejęto również bardzo użyteczny mechanizm powiązań symbolicznych (ang. *symbolic links*): można tworzyć pliki zawierające ścieżki do innych plików czy katalogów, a operację otwarcia takiego pliku system realizuje jako operację otwarcia pliku wskazywanego ścieżką, nawet jeśli wskazywany plik należy do innego systemu plików.

Pamięć wirtualna. Do pojawienia się 3BSD UNIX-a (koniec lat siedemdziesiątych) podstawową strategią zarządzania pamięcią w systemach UNIX-owych pozostawało wymiatanie (ang. *swapping*), czyli usuwanie na dysk całych obrazów procesów. Obecnie, gdy tylko pozwala na to sprzęt, stosuje się stronicowanie na żądanie (ang. *demand paging*), ewentualnie wspomagane wymiataniem w sytuacjach nietypowych. Dotyczy to również UNIX-a z AT&T począwszy od nowszych wersji SVR2. W SVR4 przejęto i udoskonalono organizację pamięci wirtualnej z SunOS (dla którego z kolei pierwowzorem były wersje 4.2BSD i 4.3BSD). Istotną nowością tego rozwiązania jest możliwość odwzorowywania plików w pamięci, w przestrzeni adresowej użytkownika (ang. *memory mapped files*), co oczywiście znacznie przyspiesza dostęp. Innym efektem przyjętej organizacji pamięci jest sprawny mechanizm współdzielenia bibliotek.

Mechanizm strumieni. Strumienie (ang. *streams*) pojawiły się w SVR3 jako sposób na przezwyciężenie niedogodności związanych z używaniem tradycyjnych UNIX-owych mechanizmów znakowego wejścia-wyjścia w oprogramowaniu protokołów sieciowych. Mechanizmy te (*c-listy* i *dyscypliny wierszowe* wykorzystywane w programach obsługi urządzeń znakowych) wymuszały nieefektywne przetwarzanie znak po znaku, utrudniały warstwowe nakładanie i dynamiczną wymianę protokołów oraz często prowadziły do powielania kodu. Strumień jest rodzajem dwukierunkowego potoku, w którym moduły przetwarzają i przekazują sobie wzajemnie buforowane dane (od programu użytkowego do programu obsługi urządzenia i odwrotnie). Programista może moduły dynamicznie dołączać i usuwać, tworząc w ten sposób rodzaj stosu aktywnych, niezależnie szeregowanych jednostek programowych. W SVR4 z ulepszonego tu mechanizmu strumieni po raz pierwszy korzysta również podsystem obsługi terminali oraz oprogramowanie potoków (ang. *pipes*).

Czas rzeczywisty. O tym, że tradycyjny UNIX nie nadaje się do zastosowań wymagających szybkiej reakcji w czasie rzeczywistym, wiadomo od dawna. Twórcy UNIX-a nie planowali takich zastosowań dla swego systemu, co wyraźnie widać w architekturze jądra – system nie wywłaszcza procesów w trakcie wykonywania przez nie funkcji systemowych, programista praktycznie nie ma kontroli nad priorytetami procesów, brak asynchronicznych operacji wejścia-wyjścia, itp. W efekcie nie da się przewidzieć czasu reakcji systemu na zdarzenia zewnętrzne. Wiadomo również – dowodem systemy *VENIX* czy *RTU* – że stosunkowo niewielkie modyfikacje jądra pozwalają usuwać większość takich ograniczeń (co nie znaczy, że modyfikowany w ten sposób UNIX mógłby szybkością dorównać specjalizowanym systemom czasu rzeczywistego, jak choćby *VRTX*). W SVR4 umożliwiono wywłaszczanie procesów w trybie jądra, programiście dano realną możliwość kontrolowania priorytetów, rozszerzono mechanizmy ochrony procesów przed wymiataniem. W przyszłości przewiduje się zgodność SVR4 ze standardem rozszerzeń czasu rzeczywistego opracowywanym przez POSIX.

Dynamiczne łączenie. Mechanizm dynamicznego łączenia (ang. *dynamic linking*), w SVR4 przyjęty z SunOS 4.0, pozwala programom korzystać z funkcji przechowywanych w skompilowanych wcześniej bibliotekach. Dynamiczne dołączanie funkcji (tylko tych rzeczywiście wykorzystywanych) ze współdzielonych bibliotek prowadzi oczywiście do mniejszych rozmiarów i większej elastyczności programów. Jedyną wadą jest nieco dłuższy czas ładowania programu. W SVR3.2 istniał wprawdzie mechanizm współdzielenia bibliotek, lecz wynikające ze statycznego łączenia uciążliwości mocno odstręczały programistów.

Standard binariów. Przenośność oprogramowania – systemowego i użytkowego – od dawna uważa się za jedną z ważniejszych zalet UNIX-a. Rzecz w tym, że UNIX-owców zawsze dotąd interesowała wyłącznie przenośność kodu źródłowego. Wiadomo jednak, że rozpowszechnianie programów użytkowych w postaci źródłowej ma swoje wady: zainstalowanie pakietu zwykle wymaga wtedy pewnej wiedzy (prawdziwie przenośny kod trudno napisać) oraz środków (translator, przestrzeń dyskowa, itp.) i, co ważniejsze, taka forma rozpowszechniania oznacza utratę kontroli nad produktem. SVR4 jest pierwszą wersją UNIX-a, która umożliwia przenoszenie programów w postaci w pełni skompilowanego kodu binarnego. Taka przenośność ogranicza się oczywiście do systemów z tym samym typem procesora (OSF planuje wprowadzić nieograniczoną przenośność w formie półskomplikowanego kodu pośredniego). Warunkiem jest stosowanie się do reguł zawartych w dokumencie definiującym sprzęg binarny

(ang. *Application Binary Interface*, ABI) programu użytkowego z systemem. Realizację przenośności umożliwia mechanizm dynamicznego łączenia bibliotek: w czasie uruchamiania programu wybiera się warianty bibliotek ABI, właściwe dla danego systemu i konfiguracji, i z nimi łączy się ładowany program.

Internacjonalizacja. Przez wiele lat UNIX ignorował potrzeby językowe i przyzwyczajenia swych nieamerykańskich użytkowników. Próby rozszerzania alfabetu o tak zwane znaki narodowe dodatkowo komplikowała uporczywa skłonność systemu do pomijania ósmego bitu w komunikacji ze światem zewnętrznym. Pierwszym ważnym krokiem ku wielojęzyczności UNIX-a stało się opracowanie standardu ANSI języka C. Zwykłe znaki stały się przyzwoicie ośmiobitowe, poza tym standard wprowadza znaki szerokie i wielobajtowe, niezbędne choćby Japończykom. Co ważniejsze, pojawiła się możliwość definiowania (i dynamicznego zmieniania) cech *lokalnych* (ang. *locales*) środowiska programu, a więc na przykład porządku w zbiorze znaków czy formatu czasu i daty. SVR4 udostępnia programiście kompilator pełnego ANSI C wraz z otoczeniem i dzięki temu staje się wielojęzykowy w tym sensie, że umożliwia tworzenie programów, z których mogą jednocześnie korzystać użytkownicy posługujący się różnymi zestawami cech lokalnych. Niestety reszta programów systemowych SVR4 (w tym sort!) wciąż jeszcze nie sprzyja cudzoziemcom.

Sprzęg użytkowy. Od kilku lat wielką UNIX-ową karierę robi *X Window System* – opracowany w MIT graficzny system okien, zakładający pracę w sieci. W SVR4 stanowi on podstawę (ewentualnie łącznie z ekranową wersją *PostScript*-u, gdy korzysta się z *X11/NEWS*) dla graficznego sprzęgu użytkowego (ang. *graphic user interface*, GUI) noszącego nazwę Open Look. Open Look jest standardem (tylko lokalnym – poważnym jego konkurentem jest OSF/Motif): definiuje jednolite reguły komunikowania się oprogramowania z użytkownikiem i zapewnia do tego środki – okna, ideogramy, kolory, mysz, itd. Istnieje również jego znakowy odpowiednik (FACE, *Framed Access Command Environment*) przeznaczony dla zwykłych terminali.

Inne nowości. Repertuar poleceń SVR4 jest swego rodzaju zunifikowaną sumą list poleceń 4.3BSD, SunOS, SCO XENIX-a i SVR3.2 W nielicznych przypadkach, w których unifikacja nie była możliwa, użytkownik może wybierać wariant konfiguracyjnie. Nowością jest powłoka Korna, *ksh*, rodzaj rozszerzonego połączenia *csh* i *sh* (Bourne'a), zgodna w dół z tą ostatnią. W przyszłości wszystkie polecenia zostaną doprowadzone do zgodności z odpowiednim standardem POSIX-a. Całości obrazu dopełniają liczne udoskonalenia w dziedzinie administracji i konfigurowania systemu.

Plany na przyszłość. Po raz pierwszy w historii w AT&T opublikowano szczegółowy, wieloletni plan rozwoju systemu. Przewiduje się w nim m.in. rozbudowę zabezpieczeń (przed wirusami, utratą danych, nieautoryzowanym dostępem), wprowadzenie wieloprzetwarzania i rozszerzeń sieciowych. Natomiast nie widać w planach generalnej przebudowy (czy choćby „odchudzenia") jądra systemu, co dla odmiany od dawna zapowiada OSF.

AIX

AIX (*Advanced Interactive Executive*) pojawił się w 1986 roku wraz z komputerami RT PC, pierwszymi stacjami roboczymi firmy IBM. Oparty na SVR2, zawierał liczne rozszerzenia i modyfikacje, którymi zdobył sobie wielu zwolenników (podobały się m.in. możliwość odwzorowywania plików w pamięci i dynamicznego dołączania programów obsługi urządzeń, łatwość administrowania oraz ogólna przyjazność). Przeciwnicy utrzymywali, że jest co najwyżej UNIX-opodobny. Potem pojawiły się wersje przeznaczone dla PS/2 i System/370, a ostatnio dla System/6000 – nowej rodziny stacji roboczych z procesorami typu RISC.

W możliwościach najnowszego AIX3 i SVR4 jest stosunkowo niewiele różnic (i nic dziwnego – w obu systemach wykorzystano całą najnowocześniejszą technologię UNIX-ową). Do najatrakcyjniejszych cech AIX-a należą: możliwość dynamicznego rozbudowywania jądra (zarówno o moduły systemowe, jak i o przygotowane przez użytkownika), system plików, który rejestruje historię zmian (co w razie awarii umożliwia odtworzenie poprzedniego stanu dysku) oraz wysoki stopień internacjonalizacji.

Mach

Mach nie jest jeszcze jednym wariantem UNIX-a, jest to raczej nowy pomysł na jądro UNIX-a. Na przykładzie SVR4 wyraźnie widać, co stało się z pierwotną ideą twórców UNIX-a: z koncepcją małego, spójnego jądra, realizującego w sumie kilka prostych mechanizmów. Mach – system autorstwa naukowców z uniwersytetu Carnegie-Mellon – jest swego rodzaju powrotem do UNIX-owych źródeł, lecz powrotem bardzo nowatorskim. Projektowany od początku z myślą o systemach wieloprocesorowych i rozproszonych, z cechami podejścia obiektowego, zawiera elementy nie spotykane dotąd w UNIX-ie.

Największą innowacją jest zastąpienie procesu zadaniem (ang. *task*), które może składać się z wielu współbieżnych wątków (ang. *thread*), zwanych też procesami wagi lekkiej (ang. *lightweight process*). Wszystkie wątki zadania działają we wspólnej przestrzeni adresowej i mają te same przywileje komunikacyjne. Czas przełączania wątków jest znacznie krótszy niż procesów (mniej informacji o stanie), a w dodatku wątki tego samego zadania mogą wykonywać różne procesory. Tradycyjne procesy można oczywiście symulować ograniczając się do zadań jednowątkowych.

Zadania i wątki komunikują się za pomocą kanałów zwanych portami – analogia ze sprzętem jest celowa. Porty można wiązać z obiektami, które uzyskują w ten sposób uprawnienia do odbioru wiadomości nadsyłanych przez obiekty mające uprawnienia do nadawania. Komunikacja z systemem (lub związana z akcjami systemu) odbywa się przez wyróżnione porty. Wiadomości mogą zawierać odniesienia do innych portów, co trochę przypomina adresację pośrednią.

Mach wyróżnia się również elegancką realizacją pamięci wirtualnej. Wprowadzono tu wyraźny podział na dwie warstwy: warstwa wyższa jest uniwersalna, niższa współpracuje z konkretnym sprzętem. Użytkownik ma możliwość dołączania własnych procedur zarządzania pamięcią, a więc może na przykład realizować własną, specyficzną strategię stronicowaną. Istnieje także mechanizm odwzorowywania plików w pamięci.

Mach oczywiście nie jest jedynym pomysłem na przyszłe jądro UNIX-a – podobne prace prowadzi się co najmniej w kilku ośrodkach. Rzecz w tym, że Mach już się sprawdził komercyjnie – w komputerach NeXT – i ma szanse na jeszcze większy sukces, o ile plany OSF się powiodą.

JACEK SURMA

W trzeciej części: o POSIX-ie i innych próbach standaryzacji UNIX-a.

/pomysły Kosz na śmieci

Rubryka tą w swoim zamyśle służyć ma publikacji pomysłów uznanych przez autorów za przydatne w ich codziennej działalności. Zdając sobie sprawę, że atrakcyjność i popularność tego działu zależy głównie od nadsyłanych przez Was rozwiązań, liczymy na szeroką współpracę.

Na początek chcieliśmy zaproponować prosty przykład skryptu rozwiązującego problem spędzający sen z powiek programisty: odzyskanie plików usuniętych zleceniem rm. Skrypt jest oparty na zasadzie śmietnika: zlecenie „przenosi" usuwane pliki do skorowidza śmieci w macierzystym skorowidzu użytkownika.

Zakładamy, że prywatne skrypty i programy użytkowe trzymane są w skorowidzu $HOME/bin.

dokończenie na s. 32

supra

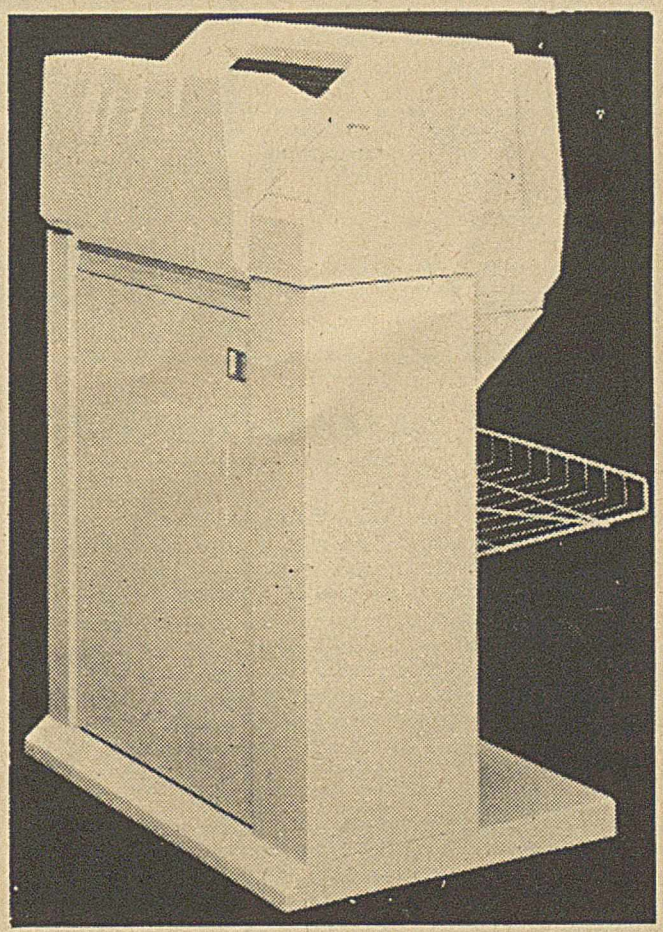
SUPRA SP. Z O.O.
ul. Trybunalska 54, 60-325 POZNAŃ
telefon: 674392, telefaks: 236498
teleks: 412966

OFERUJE ŚWIATOWY BESTSELLER

MANNESMANN
TALLY

DRUKARKĘ MOZAIKOWO-WIERSZOWĄ MT 645

- doskonała jakość druku
- niezawodność 7000 h MTBF
- duża szybkość 450 wierszy/min
- cicha praca
- bufor 27 KB
- liczba kopii 1+5
- pełna grafika 240 (dpi)
- polskie litery (opcja)
- wewnętrzny program testujący
- dwa interfejsy automatycznie przełączane
- podwójny system traktorów



MANNESMANN
TALLY

NAJPRAWDOPODOBNIEJ NAJLEPSZE
DRUKARKI NA ŚWIECIE

AUTORYZOWANY DYSTRYBUTOR DRUKAREK
MANNESMAN TALLY

supra

0/12/90

Mokrzycki W.: Wprowadzenie do maszynowego postrzegania

INFORMATYKA 1990, nr 10, s. 1

Historia badań naukowych w dziedzinie maszynowego postrzegania, matematyczne sformułowanie problemu rozpoznawania obiektów oraz charakterystyka składników systemu rozpoznającego.

Mokrzycki W.: Introduction to machine vision

INFORMATYKA 1990, No. 10, p. 1

History of scientific research in the field of machine vision, mathematical formulation of the object recognition problem and characteristics of recognition system components.

Mokrzycki W.: Einführung in maschinelle Perzeption

INFORMATYKA 1990, Nr. 10, S. 1

Geschichte der wissenschaftlichen Forschung im Bereich der maschinellen Perzeption, mathematische Formulierung des Problems von Objekterkennung und eine Charakteristik der Komponenten eines Erkennungssystems.

Górnisiewicz A., Kukuła M., Materka A.: Mikrokomputerowe systemy przetwarzania obrazów

INFORMATYKA 1990, nr 10, s. 9

Charakterystyka wyników prac badawczych z dziedziny cyfrowych metod przetwarzania obrazów prowadzonych w Instytucie Elektroniki Politechniki Łódzkiej na mikrokomputerach IBM PC, a także tendencji rozwoju mikrokomputerowych systemów przetwarzania obrazów.

Górnisiewicz A., Kukuła M., Materka A.: Microcomputer systems for image processing

INFORMATYKA 1990, No. 10, p. 9

Characteristics of scientific research results on digital image processing methods in the Electronics Institute of Łódź Technical University on IBM PC microcomputers, as well as of microcomputer image processing development trends.

Górnisiewicz A., Kukuła M., Materka A.: Mikrorechnerbildverarbeitungssysteme

INFORMATYKA 1990, Nr. 10, S. 9

Eine Charakteristik von Ergebnissen der wissenschaftlichen Forschung im Bereich von digitalen Methoden der Bilderkennung, die im Institut für Elektronik der Technischen Universität in Łódź auf IBM PC geführt werden, sowie von Entwicklungstendenzen der Mikrorechnerbildverarbeitungssysteme.

Nowiński W.L.: Problemy współbieżnej rekonstrukcji obrazów

INFORMATYKA 1990, nr 10, s. 19

Charakterystyka współbieżnej rekonstrukcji obrazów, zawierająca omówienie związanych z nią problemów języka współbieżnego programowania, współbieżnych algorytmów oraz współbieżnych architektur sprzęt komputerowego.

Nowiński W. L.: Problems of concurrent image reconstruction

INFORMATYKA 1990, No. 10 p. 19

Characteristics of concurrent image reconstruction, which includes discussion of problems related to concurrent programming languages, concurrent algorithms and concurrent hardware architectures.

Nowiński W. L.: Probleme der parallelen Bildrekonstruktion

INFORMATYKA 1990, Nr. 10, S. 19

Eine Charakteristik der parallelen Bildrekonstruktion, die eine Besprechung der mit ihr verbundenen Problemen der Sprachen von paralleler Programmierung, parallelen Algorithmen und paralleler Architektur der Hardware umfasst.

Grabska E., Hliniak G., Lembas J.: Zastosowanie grafów kompozycyjnych do konstrukcji strukturalnego edytora graficznego

INFORMATYKA 1990, nr 10, s. 23

Charakterystyka konstrukcji graficznego edytora diagramów BFE oraz możliwości zastosowania teorii grafów kompozycyjnych do opisu struktur.

Grabska E., Hliniak G., Lembas J.: Application of compositional graphs for construction of structural graphic editor

INFORMATYKA 1990, No. 10, p. 23

Charakteristics of the BFE structural graphic editor construction and application possibility of the compositional graph theory for description of structures.

Grabska E., Hliniak G., Lembas J.: Anwendung von Kompositionsgraphen zur Konstruktion des strukturellen Graphikeditors

INFORMATYKA 1990, Nr. 10., S. 23

Eine Charakteristik von Konstruktion des BTE, eines Graphikeditors für Diagramme, und von Anwendungsmöglichkeiten der Kompositionsgraphentheorie zur Beschreibung der Strukturen.

Jak startuje UNIX

dokończenie ze s. 28

Procesy te realizują program /etc/getty. Program getty, który znowu, tak jak init, możemy napisać sami, ma za zadanie wysterować przydzieloną mu linię komunikacyjną. Najpierw próbuje, na podstawie danych zapisanych w bazie danych, ustawić wstępnie parametry linii funkcją systemową ioctl( ). Następnie wysyła na terminal przyłączony do tej linii napis będący zwykle zachętą dla użytkownika do rozpoczęcia sesji i odczytuje odpowiedź użytkownika. Jeśli któraś z tych czynności nie udaje się (funkcje czytania lub pisania wracają z błędem) próbuje zmodyfikować parametry linii i rozpocząć od nowa. Jeśli znowu kończy się to niepowodzeniem, proces kończy działalność.

Po odczytaniu odpowiedzi użytkownika proces kończy realizację programu getty i zamiast niego zaczyna wykonywać (funkcją systemową exec( )) program /bin/login, przekazując mu otrzymaną odpowiedź użytkownika. Program longin zajmuje się sprawami związanymi z dopuszczeniem użytkownika do sesji i, jeśli wszystko pójdzie dobrze, ustępuje miejsca (cały czas w ramach tego samego procesu) programowi obsługi sesji użytkownika. Jest to zwykle któryś z interpretatorów poleceń (*sh, csh, vsh* itp.), choć znów może to być dowolny inny program. Omawianie tych zagadnień przekracza już jednak ramy tego artykułu.

A skądże to, jakże to, czemu tak gna?

Pozostaje jeszcze odpowiedzieć na pytanie, dlaczego po zakończeniu sesji (koniec działania procesu, który gościł kolejno programy getty, longin i program obsługi sesji) terminal nie pozostaje martwy. Dba o to init. Po otrzymaniu informacji, że jego proces potomny umarł, powołuje na nowo dla tej linii komunikacyjnej proces realizujący program getty. W ten sposób cykl się zamyka

i pociąg się toczy,
Bo para te tłoki wciąż tłoczy i tłoczy,
I koła turkocą, i puka, i stuka to:
Tak to to, tak to to, tak to to, tak to to!...

DARIUSZ KUPIECKI

/pomysły

dokończenie ze s. 30

W pliku „.profile" w skorowidzu macierzystym zmieniamy wiersz definiujący zmienną środowiskową PATH na

```
PATH=$HOME/bin:/bin:/usr/bin:.
```

i dodajemy wiersz

```
[ -d śmieci ] || mkdir śmieci
```

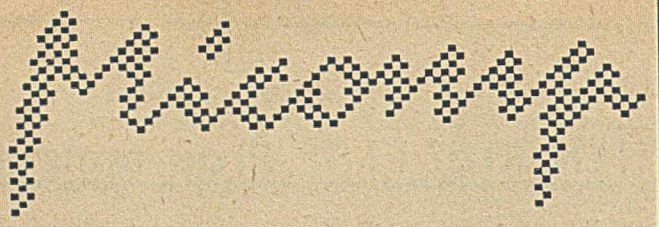
W skorowidzu $HOME/bin umieszczamy plik rm o treści

```
[ -d $HOME/śmieci ] || {
  echo rm: Nie ma skorowidza $HOME/śmieci && exit 1
  }
for i
do
  {cp $i $HOME/śmieci 2> /dev/null && /bin/rm $i
  } || echo rm: Nie moge wysłać $i do $HOME/śmieci
done
exit 0
```

Chcieliśmy zaznaczyć, że zamieszczony skrypt jest tylko szkicem rozwiązania postawionego problemu i może być wzbogacany na różne sposoby.

WŁADYSŁAW MAJERSKI

06/1181 ł
POLITECHNIKA ŚLĄSKA



ZAKŁAD SYSTEMÓW MIKROKOMPUTEROWYCH
MiCOMP

40–045 Katowice, ul. Astrów 7
telefon 513-086, telefaks 518-628,
teleks 315687

Systemy teletransmisji danych
ICL/ODRA

Integracja systemów ICL i ODRA
z mikrokomputerami standardu PC
i systemami NOVELL, UNIX SCO

procesor sieci rozległej WAN
MiCOMP 8075
(emulacja ICL 7503)

most NOVELL – ICL/ODRA
NetCROSS-FXBM

program teletransmisji danych
MiCROS-FXBM (PC, rozproszone bazy danych, wersja sieciowa – NOVELL),
skaner MPXSCAN-8007,

adaptery, modemy, testery,
emulatory terminali.

Instalacje w największych systemach
na terenie całego kraju!

E0/1124/89

Zespół elektroników

Krakowskiego
Przedsiębiorstwa
Informatyki
Przemysłu
Budowlanego
ETOB

wykonuje

z własnych materiałów

sterowniki

pamięci

taśmowych

do komputera Odra 1305
w technice TTL
(funkcjonalny odpowiednik
MTS–304–2M/K).

Konkurencyjnie niskie ceny!

Informacji udziela
i zamówienia przyjmuje:

KPIPB „ETOB"
30–415 Kraków, ul. Wadowicka 10
tel. 66-69-40 lub 66-80-22 w. 124

0/15/90